MATLAB 图形技术

——绘图及图形用户接口

周明　李长虹　雷虎民　编著

西 北 工 业 大 学 出 版 社

1999 年 11 月　西安

（陕）新登字 009 号

【内容简介】 本书主要介绍基于MATLAB的图形设计技术。全书主要由两大部分组成：二维图形及三维图形的绘制技术；图形用户接口设计技术。为提高本书的实用性，在附录中介绍了MATLAB 5.X版所提供的20多类基本命令函数，以便用户编制程序时参考。

本书既可作为大专院校师生的指导书，也可作为科研及工程技术人员高效、实用的工具书。

图书在版编目（CIP）数据

MATLAB图形技术：绘图及图形用户接口/周明，李长虹，雷虎民编著．—西安：西北工业大学出版社，1999.11

ISBN 7-5612-1192-9

Ⅰ．M… Ⅱ．①周… ②李… ③雷… Ⅲ．①计算机辅助计算-软件包，MATLAB-图形软件-程序设计 ②计算机辅助计算-软件包，MATLAB-图形软件-接口-程序设计 Ⅳ．TP391.75

中国版本图书馆CIP数据核字（1999）第51561号

*

（邮编：710072 西安市友谊西路127号 电话：8493844）

全国各地新华书店经销

西安市向阳印刷厂印装

*

开本：787毫米×1 092毫米 1/16 印张：14.375 字数：343千字

1999年11月第1版 1999年11月第1次印刷

印数：1—4 000册 定价：22.00元

前　　言

MATLAB 语言被称为是一种"演草纸式的科学计算语言"，它在数值计算、数据处理、自动控制、信号处理、神经网络、优化计算、模糊逻辑、小波分析、图像处理、统计分析、金融分析等众多的领域有着广泛的用途。特别是它所提供的各种工具箱，使得我们在科学计算、工程设计、数值分析等领域中的各种计算、演算、模拟等工作变得相当简单。通过几条简单的 MATLAB 命令，就可完成一大串高级计算机语言才能完成的任务，它不仅具有面向对象的计算机语言特征，也初具面向任务的计算机语言的思想。这就是 MATLAB 魅力之所在。

图形显示技术使得我们可以简单地理解问题、分析问题。人们在现实生活中所接受的信息，绝大多数都是由视觉器官所获取的。图形技术使我们不用再去观察繁琐无味的数字，图形技术能使我们更好地理解数字。但遗憾的是在计算机上绘制图形却并不是一件很简单的事。而 MATLAB 所提供的各种图形设计技术，使得我们不用再去过多地考虑图形实现技术的细节内容，它所提供的多种图形命令使得图形显示技术变得简单易行，甚至用一条命令就可得到直观、形象的图形结果。

目前，Windows 系统已成为微机操作系统的主流，Windows 系统的发展也推动着微机操作技术的进步。MATLAB 语言所提供 GUI(图形用户接口)设计技术使得我们可以方便地设计出与 Windows 系统相类似的用户接口，使得我们所开发出的各种应用程序、应用系统更为众多的用户所接受。

本书主要介绍基于 MATLAB 的图形设计技术。全书主要由两大部分组成：二维图形及三维图形的绘制技术；图形用户接口设计技术。全书共分为如下 5 章及 1 个附录：

第 1 章作为本书的基础，主要介绍 MATLAB 语言的主要思想精要，并简单地介绍了 MATLAB 语言的程序设计基本技术。

第 2 章介绍了二维图形绘制技术，包括基本二维图形绘制技术以及一些常用的特殊二维图形绘制方法。

第 3 章介绍了三维图形绘制技术，包括三维图形的基本绘制技术以及一些常用的特殊三维图形绘制方法。

第 4 章详细介绍了 MATLAB 所提供的 10 个图形对象及其各种属性，它使得我们可以把握图形设计的细节技术。

第 5 章介绍了基于 MATLAB 的图形用户接口设计技术，包括各种控件的实现方法。

在附录中，为提高本书的实用性，我们简单地介绍了 MATLAB 5.X 版所提供的 20 多类基本命令函数，以便用户编制程序时参考。

本书为“MATLAB 语言系列丛书”之三，有关 MATLAB 语言的基础知识（软件安装、基本使用方法、编程技术）及动态仿真工具 SIMULINK 的技术、技巧知识，读者可阅读本丛书之一——《MATLAB 语言精要及动态仿真工具 SIMULINK》（定价 18.00）；关于 MATLAB 五大工具箱（控制、信号处理、优化、神经网络、模糊推理）的介绍与使用，请参阅本丛书之二 ——《MATLAB 语言工具箱 ——TOOLBOX 实用指南》（定价 28.00 元）。

本书由周明、李长虹、雷虎民共同编著。全书由周明统一定稿。

由于作者水平有限，本书又为全国第一本全面介绍 MATLAB 图形技术的实用图书，在写作过程中难免有把握不准及遗、讹之处，敬请读者多提宝贵意见，以便继续完善，共同进步。

作　者

1999 年 5 月

目　　录

第1章 MATLAB语言程序设计精要

MATLAB语言被称为一种“演草纸式的科学计算语言”,它在数值计算、数据处理、自动控制、信号处理、神经网络、优化计算、模糊逻辑、小波分析、图像处理、统计分析等众多的领域有着广泛的用途,也对这些学科的研究和发展起着重要的作用。作为本书的语言基础,我们在本章简要地介绍MATLAB语言的思想精华,并介绍MATLAB语言的一些基本语句,以及MATLAB语言的基本编程方法。本章的内容是继续阅读本书的重要基础,当然,熟悉MATLAB语言基本内容的读者也可跳过本章。本章具体包括以下内容:

- MATLAB语言简介
- MATLAB的工作环境
- MATLAB的基本操作方法
- MATLAB中的常用数学函数
- MATLAB中的关系运算与逻辑运算
- MATLAB中的矩阵运算方法
- MATLAB语言中的矩阵函数
- MATLAB中的字符串及其处理函数
- MATLAB语言中的数值分析计算
- MATLAB语言的控制语句
- MATLAB语言的辅助语句
- MATLAB语言的输入及输出语句
- MATLAB语言的文件操作方法
- MATLAB语言中的用户自定义函数

1.1 MATLAB语言简介

有一点使用过计算机语言编程经验的读者可能都会有这样的体会,当我们进行程序设计时,特别是当程序涉及到矩阵运算或绘图时,程序的编制过程是比较繁琐的,尤其是当我们需要编制出一个通用程度较高的程序时就更为麻烦。它不仅要求我们深刻了解所要求解的问题以找到一个可靠性较好的算法,还必须研究各种可能的边界条件,特别是要考虑各种范围的数据大小等,另外,还要熟练掌握所使用的计算机语言。即便如此,所编写出的程序仍有可能会由于这样或那样的原因出错,或得不到满意的结果。

矩阵运算是很多领域都可能涉及到的一种数学方法。显然,当涉及到矩阵运算时(如矩阵求逆、矩阵相乘、求矩阵的特征值等),如果都由研究人员或工作人员亲自使用某种高级语言来

编写一个通用性较好的程序，肯定是一种既复杂又浪费精力且意义不大的工作。有感于此，美国的 Cleve Moler 博士开发出了一套交互式的软件系统，称为 MATLAB(MATrix LABoratory，即矩阵实验室)。在这个 MATLAB 环境下，矩阵的运算变得非常简单。在此基础上，Moler 博士等一批数学家与一些软件专家联合组建了一个名为 MathWorks 的软件开发公司，专门扩展并改进 MATLAB。1992 年，MathWorks 公司推出了 MATLAB 4.0 版，随后推出了可应用于 Windows 上的微机版，1997 年又推出了功能更加完善的 5.1 版。

目前，MATLAB 已不再仅仅只是一个"矩阵实验室"，经过不断的发展，MATLAB 增加了很多不同用途的工具箱，这些工具箱大多由某一领域的名家所开发，从而使得 MATLAB 已经成为了国际上最流行的一种具有广泛应用前景的计算机高级编程语言，并赢得了"演草纸式的科学计算语言"的美称，由此也可见其操作与编程的简单性。现在，国外一些大学已经把 MATLAB 语言列为大学生应掌握的一种基本工具。

从目前 MATLAB 的性能来讲，总体上说，与其他一些高级计算机语言相比，用 MATLAB 语言编写的程序的执行效率要低，但其编程效率、程序的可读性、可移植性都要远远高于其他高级语言。

从功能上来说，MATLAB 不仅提供了丰富可靠的矩阵运算函数，还增加了图形图像处理功能、声音处理功能、数据处理功能、Windows 图形用户界面设计功能等，特别是还开发出了多种以 MATLAB 为基础的实用工具箱，从而使得 MATLAB 可以广泛地应用于自动控制、图像处理、信息处理、生物医学工程、语言处理、雷达工程、振动分析、时间序列分析与建模、优化计算等领域。较为常用的 MATLAB 工具箱主要包括：

(1) 控制系统工具箱；
(2) 系统辨识工具箱；
(3) 鲁棒控制工具箱；
(4) 多变量频率设计工具箱；
(5) μ 分析与综合工具箱；
(6) 神经网络工具箱；
(7) 最优化工具箱；
(8) 信号处理工具箱；
(9) 模糊推理系统工具箱；
(10) 小波分析工具箱；
(11) 样条函数工具箱；
(12) 符号数学工具箱；
(13) 统计工具箱；
(14) 图像处理工具箱；
(15) 微分方程工具箱；
(16) 金融工具箱；
(17) 高阶频谱分析工具箱；
(18) LMI 控制工具箱；
(19) QFT 控制工具箱。

1.2 MATLAB 的工作环境

1.2.1 启动 MATLAB

在计算机上安装好 MATLAB 之后，双击 MATLAB 图标，或者是通过 Windows 的运行命令执行 MATLAB 的启动程序 matlab.exe，即可进入 MATLAB，此时系统将显示出 MATLAB 的命令窗口，如图 1－1 所示。

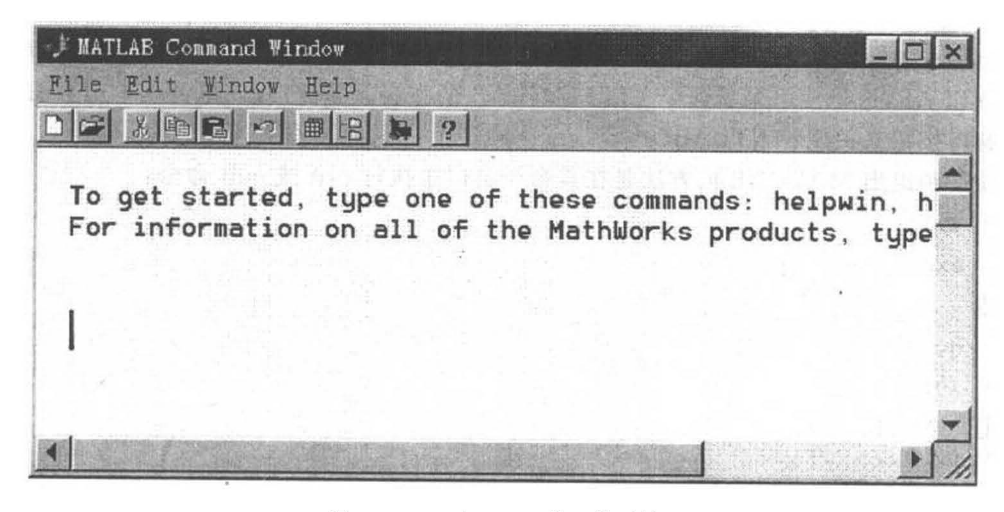


图 1－1 MATLAB 的工作环境

在图 1－1 中，闪烁的光标就表示 MATLAB 的命令提示符（中文版 MATLAB 的命令提示符为"》"）。在这个提示符之后，我们可以输入各种操作命令，让 MATLAB 执行某种工作。

MATLAB 给我们提供的主要菜单命令 如下。

一、File 子菜单

(1) New 菜单命令　打开一个新的 MATLAB 编辑文件、图形文件或模型文件。

(2) Open 菜单命令　打开一个已存在的 MATLAB 文件或模型文件。

(3) Load Workspace 菜单命令　调入一个已保存的 MATLAB 工作空间。

(4) Save Workspace As 菜单命令　将当前 MATLAB 工作空间的变量保存到一个文件。

(5) Show Workspace 菜单命令　显示当前 MATLAB 工作空间中的变量。

(6) Show Graphics Property Editor 菜单命令　显示图形属性编辑器。

(7) Show GUI Layout Tool 菜单命令　显示图形用户接口(GUI)布置工具。

(8) Preferences 菜单命令　设置 MATLAB 的各种系统属性。

(9) Print 菜单命令　打印当前窗口的内容。

(10) Exit MATLAB 菜单命令　退出 MATLAB。

二、Edit 菜单

(1) Undo 菜单命令　撤消上次编辑操作。

(2) Cut 菜单命令　将选定的内容剪裁到粘贴版。

(3) Copy 菜单命令　将选定的内容复制到粘贴版。

(4) Paste 菜单命令　将粘贴版中的当前内容粘贴到光标处。

(5) Clear 菜单命令　清除所选定的内容。

三、Window 菜单

用于切换 MATLAB 的各个工作窗口。

四、Help 菜单

用于显示 MATLAB 的各种帮助信息。

1.2.2　退出 MATLAB

当用户执行完 MATLAB 的程序或命令之后，可由"File"→"Exit MATLAB"菜单命令退出 MATLAB，这是使用菜单命令关闭 MATLAB 的方法。当然，我们也可用单击其右上角的关闭窗口按钮来关闭 MATLAB。

另一种退出 MATLAB 的方法是在其命令窗口中执行 exit 或 quit 命令：

· 格式：

exit

或

quit

· 功能：

退出 MATLAB。

1.3　MATLAB 的基本操作

1.3.1　变量及其命名规则

我们可以把 MATLAB 语言看作是一种解释性的计算机语言，用户可以在 MATLAB 的工作空间中键入某一个或一些命令，也可以应用 MATLAB 语言编写一些应用程序，在执行这些命令或程序时，MATLAB 的系统软件对这些命令或程序中的各条语句进行翻译，然后在 MATLAB 环境中对它们进行处理，并返回相应的运算结果。

MATLAB 语言是由专门用于矩阵运算的计算机语言发展起来的，其最重要的功能就是对实数矩阵或复数矩阵进行与矩阵相关的一些运算，这也是 MATLAB 的一个最基本的功能。矩阵是 MATLAB 语言的基本运算对象。另外，向量可作为矩阵的一列或一行，标量也可看作是只含有一个元素的矩阵，所以向量和标量也都可作为一种特殊的矩阵来处理。

与其他计算机语言一样，MATLAB 也可根据需要对其工作空间中所存放的数据命名，以便于进一步的运用。MATLAB 语言有下述几条变量命名规则：

(1) 变量名的首字符必须是字母。

(2) 变量名中可以有数字、下划线，但不能有标点符号。

(3) 每个变量名最长只能包含 19 个字符。

(4) MATLAB 区分变量名的大小写。如 xy，Xy，xY，XY 在 MATLAB 中分别代表 4 个不同的变量名。

1.3.2 MATLAB 的赋值语句结构

MATLAB 以复数矩阵为最基本的运算单元，它既可以对矩阵的整体进行运算处理(如矩阵求逆等)，也可以对矩阵的某一个或某一些元素进行操作。对矩阵或其元素进行赋值是 MATLAB 中的一条最基本的语句，其结构如下：

· 格式：

variable = expressions

· 功能：

将表达式 expressions 的值赋给变量 variable。

· 说明：

① 等号左边的"variable"表示变量名称。若省略该变量名，表示将指定表达式的值赋给 MATLAB 所定义的一个保留变量 ans。例如：

```
》1 + 2 + 3
ans =
          6
```

② 等号右边的"expressions"既可以是 MATLAB 所能接受的一个表达式，也可以是一个函数。

③ 若整个赋值语句用分号";"结束时，MATLAB 将不在屏幕上显示该语句的运行结果。例如：

```
》x = 111 ;
》y = 222
y =
        222
```

④ 几条 MATLAB 语句可以出现在同一行中，只是要用分号或逗号将每条语句分割开来。例如：

```
》x1 = 5 , x2 = 10 , x3 = 15
x1 =
        5
x2 =
        10
x3 =
        15
```

⑤ 若等号右边的表达式太长，可由 3 个点号"…"(称为续行符号)来结束本行，然后再在下一行继续输入表达式。例如：

```
》sum =1 + 2 + 3 + 4 + 5 + ...
        6 + 7 + 8 + 9 + 10
sum =
        55
```

⑥ MATLAB 还定义了几个特殊的常量，它们也可出现在MATLAB表达式中。这些常量主要有：

eps：表示一个很小的数，$eps=2.2204\times10^{-16}$

pi：表示圆周率 π

Inf：表示无穷大 ∞

NaN：表示一个不确定的数(Not a Number)

我们可由下述命令来了解当前已经在MATLAB工作空间中给哪些变量赋过值，即目前已定义了哪些变量。

- 格式：

who

- 功能：

显示当前已定义的变量名称。例如：

```
»who
Your variables are:
x     sum     x2
y     x1      x3
```

- 格式：

whos

- 功能：

详细显示当前已定义的多个变量的名称、维数、字节长度及类型等信息。例如：

```
»whos
  Name     Size      Bytes Class
  x        1x1       8 double array
  y        1x1       8 double array
  sum      1x1       8 double array
  x1       1x1       8 double array
  x2       1x1       8 double array
  x3       1x1       8 double array
Grand total is 6 elements using 48 bytes
```

1.3.3 数据的显示格式

一般情况下，当我们对变量进行赋值或执行MATLAB语句时，其执行结果都可在屏幕上显示出来。虽然MATLAB总是以双精度来存储数据并执行所有算术运算的，但其运算结果却可以有短格式、长格式、分数格式等多种屏幕显示格式。在MATLAB环境中，我们通过“File”→“Preferences”菜单命令所弹出的对话框，在其“General”页面中就可确定选用哪种数据显示格式，如图1-2所示。在选定了某种显示格式后，此后的所有数字输出结果都采用此格式，除非重新更改为另一种显示格式。需要说明的是，显示格式不影响数据的存储和运算精度。

我们也可通过 MATLAB 的 format 命令来设置显示格式，该命令的用法见表 1－1。

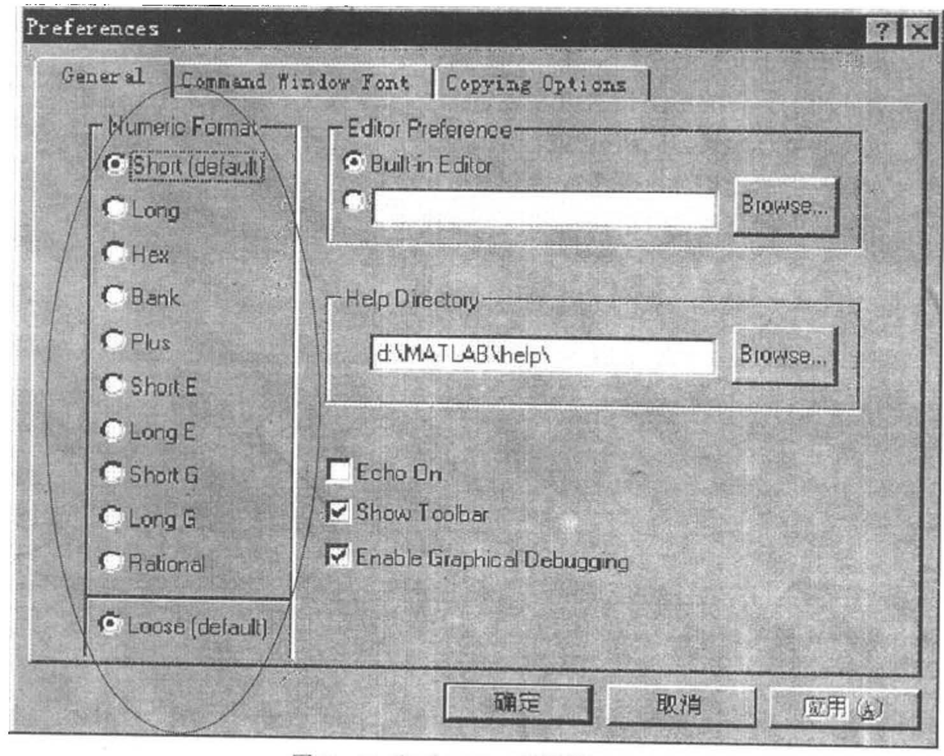


图 1－2 Preferences 对话框

表 1－1 format 命令的使用方法及示例

命令格式	含义	数据 99.123456789 的显示示例
format	默认格式(短格式)	99.1235
format short	短格式(5 位定点数)	99.1235
format long	长格式(15 位定点数)	99.12345678900000
format short e	短格式 e 方式	9.9123e+001
format long e	长格式 e 方式	9.912345678900000e+001
format short g	短格式 g 方式	99.123
format long g	长格式 g 方式	99.123456789
format hex	十六进制格式	4058c7e6b74dce59
format +	紧密格式	+
format bank	银行格式	99.12
format rat	有理格式	8029/81

1.3.4 矩阵的生成

矩阵是 MATLAB 最基本的处理单元，并且，一个标量可看做是只含有一个元素的矩阵，

而向量也可看做是只含有一列或一行的矩阵。如何产生矩阵是MATLAB的一个常用的操作。

在MATLAB中输入矩阵时有下面几条要求：

(1) 矩阵的各个元素必须包括在一对方括号中；

(2) 矩阵同一行中的各个元素用逗号分隔；

(3) 矩阵的每行元素之间用分号分隔；

(4) 方括号中的逗号可由一个或多个空格代替；

(5) 方括号中的分号也可由回车换行符号代替。

例如，矩阵 $\boldsymbol{A}=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$ 可按下列形式输入：

```
》A = [ 1,2,3;4,5,6;7,8,9 ]
A =
        1   2   3
        4   5   6
        7   8   9
```

这个矩阵 **A** 也可按如下方式输入：

```
》A = [ 1   2   3
        4   5   6
        7   8   9 ]
```

矩阵中的各个元素不仅可以是常量，也可以是一个MATLAB表达式。例如：

```
》B = [ 10,11,6 * 2 ]
B =
       10   11   12
```

矩阵的各个元素也可以是复数。在MATLAB中，复数单位表示为i=sqrt(−1)或j=sqrt(−1)，其值在工作空间中都显示为:0+1.0000 i。复数可由下面的语句生成：

$$z = a + b * i \quad (\text{简写为} \quad z = a + bi)$$

或

$$z = r * \exp(i * \alpha) \quad (\text{简写为} \quad z = r * \exp(\alpha i))$$

其中 α 为复数幅角的弧度数，r为复数的模。

MATLAB提供了两个方便的方法来输入复数矩阵，如

$$Z = [\,11,\ 22\ ;33,\ 44] + i * [\,55,\ 66\ ;77,88\,]$$

或

$$Z = [\,11+55i,\ 22+66i\ ;33+77i,44+88i\,]$$

两式具有相同的结果。

```
》Z = [ 11, 22 ;33, 44] + i*[ 55, 66 ;77, 88 ]
Z =
    1.0000 + 5.0000i   2.0000 + 6.0000i
    3.0000 + 7.0000i   4.0000 + 8.0000i
》Z = [ 11+55i ,22+66i ;33+77i ,44+88i ]
    Z =
        1.0000 + 5.0000i   2.0000 + 6.0000i
        3.0000 + 7.0000i   4.0000 + 8.0000i
```

要注意的是，复数在书写时，其中间不能有空格。例如，对于复数 1+5 * i，若在其中的加号前有空格的话（即 1 +5 * i），就会被 MATLAB 认为是两个分开的数：1 及 0+5 * i。

另外，矩阵元素还可以是另一个矩阵，但对这种写法的要求是它必须能够产生一个合法的矩阵。例如：

```
》C = [ A;B ]
C =
      1    2    3
      4    5    6
      7    8    9
     10   11   12
```

即上述矩阵 ***C*** 的前 3 行是由矩阵 ***A*** 构成的，最后 1 行是由矩阵 ***B*** 构成的。

矩阵中的每一个元素可由其下标来表示。例如，若已知

D = [11 22 33 44 55 66]

则矩阵 ***D*** 的第 1 个元素为 D(1)=11，第 2 个元素为 D(2)=22，…，第 6 个元素为 D(6)=66。若已知

$$\boldsymbol{A}=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

则　　　A(1,1)=1，A(1,2)=2，…，A(3,3)=9

矩阵中的元素可以直接依据其下标进行赋值运算，例如我们可以将上述矩阵 ***A*** 的第 A(3,3)个元素设定为 99。

```
》A(3,3) = 99
A =
      1   2   3
      4   5   6
      7   8  99
```

在计算过程中，有时我们须要知道一个矩阵的维数大小。矩阵的维数大小可由 size( )函数来查询，其用法如下：

- 格式：

size(X)

· 功能：

查询矩阵 X 的维数。

例如：

```
》E = [1  2  3  4  5  6  7  8];
》m = size(E)
m =
      1  8
```

1.3.5 特殊矩阵的生成

在科学计算及工程计算中，有时我们会用到一些特殊的矩阵，如零矩阵、单位矩阵、空矩阵等。与上述产生一般矩阵的方法不同，我们可以用更加简洁的方法来产生这些特殊矩阵。

一、零矩阵

· 格式：

variable = zeros(n)

· 功能：

产生 1 个维数为 n×n 的元素全部为 0 的零矩阵。

例如：

```
》ZeroMatrix = zeros(3)
ZeroMatrix =
              0  0  0
              0  0  0
              0  0  0
```

二、元素全为 1 的矩阵

· 格式：

variable = ones(n)

· 功能：

产生 1 个维数为 n×n 的元素全部为 1 的矩阵。

例如：

```
》OneMatrix = ones(3)
OneMatrix =
              1  1  1
              1  1  1
              1  1  1
```

三、单位矩阵

· 格式：

variable = eye(n)

· 功能：

产生 1 个维数为 n×n 的单位矩阵。

例如：

```
》EyeMatrix = eye(3)
EyeMatrix =
                1  0  0
                0  1  0
                0  0  1
```

四、空矩阵

- 格式：

  variable = [ ]

- 功能：

产生 1 个维数为 0×0 的矩阵，称为空矩阵。

空矩阵的引入，主要是为了避免程序出错。从数学含义上讲，空矩阵本身是空的，我们不能对其进行一般数学意义上的计算处理。但如果定义了空矩阵，一些矩阵函数，如 det( )，cond( )，prod( )，sum( )等，不管是对什么样的矩阵，都可以返回一个值，而不至于使程序在用到这些函数时出错。空矩阵存在于 MATLAB 的工作空间中，只是具有空的尺寸而已，当用 size( )函数测试表明某个矩阵为空矩阵时，用 exist( )函数可以测试出其确实存在。例如：

```
》EmptyMatrix=[ ]
EmptyMatrix =
     [ ]
》size(EmptyMatrix)
ans =
     0   0
```

五、随机矩阵

- 格式：

  variable = rand(m , n)

  variable = randn(m , n)

- 功能：

rand(m,n)函数产生 1 个维数为 m×n 的均匀分布的随机矩阵；randn(m,n)函数产生 1 个维数为 m×n 的符合正态分布的随机矩阵。

例如：

```
》RandMatrix1 = rand(2,6)
RandMatrix1 =
        0.9501   0.6068   0.8913   0.4565   0.8214   0.6154
        0.2311   0.4860   0.7621   0.0185   0.4447   0.7919
》RandMatrix2 = randn(2,6)
RandMatrix2 =
       -0.4326   0.1253  -1.1465   1.1892   0.3273  -0.1867
       -1.6656   0.2877   1.1909  -0.0376   0.1746   0.7258
```

这里顺便提一下，MATLAB 5.X 还支持多维阵列，并且这里所介绍的 zeros( )，ones( )，rand( )，randn( )等函数都能够产生超过两维的多维阵列数据。例如：

```
》Data3D = ones(2,2,2)
Data3D(:,:,1) =
     1     1
     1     1
Data3D(:,:,2) =
     1     1
     1     1
```

1.3.6 向量的构成

在 MATLAB 中，对于元素个数不太多的向量，我们可以直接以枚举的方法输入该向量。例如：

```
》V1 = [ 1,2,3,4,5 ]
V1 =
     1  2  3  4  5
》V2 = [ 1;2;3;4;5 ]
V2 =
     1
     2
     3
     4
     5
```

当元素个数较多，且各元素之间有一定规律可循时，我们可以利用 MATLAB 提供的冒号操作符“:”来产生向量。含有冒号操作符的表达式的一般形式是

start:step:end

其中，start 为起始值，step 表示步长，end 表示终止值。使用上述冒号表达式，我们可以产生一个以 start 为第 1 个元素值，以 step 为增量，并且最后 1 个元素值不超过 end 的行向量。当步长值为 1 时，可省略 step 参数。另外，步长参数 step 也可取负值。

例如，下面的操作分别产生 x，y，z 三个行向量。

```
》x = 1 : 8
x =
     1  2  3  4  5  6  7  8
》y = 8 : -1 : 1
y =
     8  7  6  5  4  3  2  1
》z = 0 : 0.25 : 1
z =
     0   0.2500   0.5000   0.7500   1.000
```

利用向量及冒号表达式，也可以很方便地产生有一定规律的、元素较多的矩阵。例如，我们

首先要用它产生行向量，然后进行转置，再利用所得的列向量计算出另一列向量，即可合成有两列的矩阵。例如：

```
》x = (0 : 0.2 : 2)';
》y = 10 * sin(x * pi);
》z = [x  y  2 * y]
z =
          0          0          0
     0.2000     5.8779    11.7557
     0.4000     9.5106    19.0211
     0.6000     9.5106    19.0211
     0.8000     5.8779    11.7557
     1.0000     0.0000     0.0000
     1.2000    -5.8779   -11.7557
     1.4000    -9.5106   -19.0211
     1.6000    -9.5106   -19.0211
     1.8000    -5.8779   -11.7557
     2.0000     0.0000     0.0000
```

（1）另一种按等间隔分布数据点产生向量的方法是利用 MATLAB 所提供的 linspace( ) 函数，该函数用法如下：

- 格式：

  variable = linspace(x1, x2, n)

- 功能：

  在 x1 和 x2 之间产生 n 个等间隔分布的数据点。

（2）MATLAB 还提供了一种按对数间隔分布数据点产生向量的函数 logspace( )，该函数的用法如下：

- 格式：

  variable = logspace(x1, x2, n)

- 功能：

  在 x1 和 x2 之间按对数间距产生 n 个数据点。

例如：

```
》v1 = linspace(10, 20, 6)
v1 =
    10   12   14   16   18   20
》v2 = logspace(10, 20, 6)
v2 =
   1.0 e+020 *
     0.0000   0.0000   0.0000   0.0001   0.0100   1.0000
```

1.3.7 子矩阵的产生

一、利用冒号操作符产生子矩阵

利用前述冒号操作符，可直接由一个矩阵产生另一个子矩阵。

假设我们已知一个矩阵 ***A***，则

A(m1:m2 , n1:n2)

可表示矩阵 ***A*** 的一个子矩阵，该子矩阵的元素分别取自矩阵 ***A*** 的第 m1 行到 m2 行，第 n1 列到第 n2 列。例如：

```
》A = [ 1,2,3;4,5,6;7,8,9 ];
》B = A(1:2 , 2:3)
B =
    2  3
    5  6
```

即矩阵 ***B*** 由矩阵 ***A*** 的第 1,2 行的第 2,3 列元素所构成。

注意，A(: , j)可表示矩阵 ***A*** 的第 j 列全部元素，***A***(i , :)可表示矩阵 ***A*** 第 i 行的全部元素。即缺省 m1,m2 就表示矩阵的全部行，缺省 n1,n2 则表示全部列。

二、利用 0－1 逻辑向量产生子矩阵

我们还可以利用从关系运算所建立的 0－1 向量为索引来产生子矩阵。假设 *A* 是一个 m×n的矩阵，***L*** 是一个 m 维的 0－1 向量，则 A(L,:)将给出 ***L*** 中非零元素所对应的 *A* 的行元素所组成的子矩阵。例如，下面的命令从矩阵 ***A*** 中取出第 2 列元素不等于 5 所对应的行来组成子矩阵 ***B***。

```
》A = [ 1  2  3
        4  5  6
        7  8  9 ];
》L = A(:,2)~=5;      % 求第 2 列元素不等于 5 的行索引值(向量)
》B=A(L,:)
B =
    1  2  3
    7  8  9
```

要注意的是，这种方法中的向量 ***L*** 必须是一个 0－1 逻辑值向量，而不能是一个 0－1 数值向量。

1.4 MATLAB 的常用数学函数

调用数学函数使得我们不需要对任何标准运算功能都重新编写程序，这样可减少编程工作量。MATLAB 提供了丰富的数学函数，用户可以根据需要很方便地调用这些函数。MATLAB 所提供的部分常用数学函数，见表 1－2。

表 1-2 MATLAB 所提供的部分常用数学函数

种类	MATLAB 数学函数	功能描述	种类	MATLAB 数学函数	功能描述
基本数学运算函数	abs(x)	求绝对值或复数的模	三角函数	sin(x)	正弦函数
	angle(x)	求复数幅角的弧度数		con(x)	余弦函数
	ceil(x)	求不小于 x 的最小正数		tan(x)	正切函数
	conj(x)	求共轭复数		asin(x)	反正弦函数
	exp(x)	指数函数		acos(x)	反余弦函数
	fix(x)	取整函数		atan(x)	反正切函数
	floor(x)	求不大于 x 的最大正数		atan2(x,y)	第四象限反正切函数
	gcd(x,y)	求 x,y 的最大公因数		sinh(x)	双曲正弦函数
	lcm(x,y)	求 x,y 的最小公倍数		cosh(x)	双曲余弦函数
	log(x)	自然对数函数		tanh(x)	双曲正切函数
	log10(x)	常用对数函数		asinh(x)	反双曲正弦函数
	imag(x)	求复数的虚部		acosh(x)	反双曲余弦函数
	real(x)	求复数的实部		atanh(x)	反双曲正切函数
	rem(x,y)	求 x/y 的余数	高等数学运算函数	bessel(x)	Bessel 函数
	round(x)	四舍五入圆整函数		beta(x)	Beta 函数
	sign(x)	符号函数		erf(x)	误差函数
	sqrt(x)	开平方函数		erfinv(x)	反误差函数
				gamma(x)	Gamma 函数

1.5 MATLAB 语言中的关系运算与逻辑运算

1.5.1 关系运算

当两个矩阵的维数相同时，我们可对它们各个元素之间的大小关系进行比较，这种比较称为关系运算。MATLAB 提供了 6 个关系运算操作符，见表 1-3。

表 1-3 关系运算操作符

关系运算操作符	注释
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	等于
~=	不等于

比较两个元素的大小关系时，若结果为真就用 1 表示，结果为假则用 0 表示。例如，2+2=5(2 加 2 等于 5)这个表达式显然为假，则其结果为 0。

```
》A = [ 11 , 22 ; 33 , 44 ];
》B = [ 10 , 20 ; 80 , 90 ];
》A >= B
ans =
        1  1
        0  0
```

1.5.2 逻辑运算

MATLAB 提供了 3 种逻辑运算，见表 1-4。

表 1-4 逻辑运算操作符

逻辑运算操作符	注　　释
&	与(and)
|	或(or)
~	非(not)

"&"和"|"操作符可分别用于对两个标量或两个维数相同的向量或矩阵进行"逻辑与"、"逻辑或"运算。当运算对象是向量或矩阵时，这种逻辑运算的具体操作对象是其中的各个元素。A&B 运算的结果是将 A 和 B 中都非零的元素所对应的位置设置为逻辑 1，否则设置为逻辑 0；而 A|B 运算的结果是将 A 或 B 中非零的元素所对应的位置设置为逻辑 1，否则设置为逻辑 0；"逻辑非"运算"~"的结果是将操作对象中非零元素所对应的位置设置为逻辑 0，零元素所对应的位置设置为逻辑 1。

要说明的是，不管运算对象是什么样的矩阵，逻辑运算的结果都是 0-1 矩阵。

例如：

```
》A = [0  1  2  3  3  2  1  0];
》B = [ 1  2  3  4  5  6  7  8];
》A & B
ans =
    0  1  1  1  1  1  1  0
》A | B
ans =
    1  1  1  1  1  1  1  1
》~ A
ans =
    1  0  0  0  0  0  0  1
```

1.5.3 常用的关系运算与逻辑运算函数

除了上面所介绍的关系运算符和逻辑运算符之外，MATLAB 还提供了下面一些常用的关系运算函数和逻辑运算函数，见表 1 - 5。

表 1 - 5 常用关系运算与逻辑运算函数

函　数	注　释
all(x)	向量 x 中的每一个元素非零时返回 1；否则返回 0
any(x)	向量 x 中的任一个元素非零时返回 1；否则返回 0
exist(x)	检查是否存在该变量。若存在，返回 1；否则返回 0
isempty(x)	检查 x 是否为一个空矩阵。若是，返回 1；否则返回 0
isglobal(x)	检查 x 是否为一个全局变量。若是，返回 1；否则返回 0
isinf(x)	当 x 是 Inf 时返回 1；否则返回 0
isnan(x)	当 x 属于(－∞ ，＋∞)时返回 1；而当 x 等于 NaN 时，返回 0
isstr(x)	检查 x 是否为一个字符串变量。若是，返回 1；否则返回 0
finite(x)	当 x 是 NaN 时返回 1；否则返回 0
xor(x,y)	异或运算

例如：

```
》A = [0  1  2  3 ; 3  2  1  0];
》any(A)
ans =
     1  1  1  1
》all(A)
ans =
     0  1  1  0
》isempty(A)
ans =
          0
```

1.6 矩阵运算

矩阵是 MATLAB 的基本处理单元，MATLAB 语言提供了丰富的矩阵运算能力。

1.6.1 矩阵的转置运算

MATLAB 语言使用在矩阵变量后加符号“ ' ”的方法来表示矩阵的转置运算。例如：

```
》A = [ 1,2,3;4,5,6;7,8,9 ];
》B = A'
B =
    1   4   7
    2   5   8
    3   6   9
```

注意,如果矩阵 **Z** 是一个复数矩阵,则 **Z'** 被定义为其复数共轭转置矩阵。例如:

```
》Z = [ 1+1i   1+2i
        2+1i   2+2i ];
》A = Z'
A =
    1.0000 - 1.0000i   2.0000 - 1.0000i
    1.0000 - 2.0000i   2.0000 - 2.0000i
```

对于一个复数矩阵 **Z**,若我们只想将其排列形式转置,则可以使用 Z.'或函数 conj(Z')来实现。例如:

```
》Z = [ 1+1i   1+2i
        2+1i   2+2i ];
》B = Z.'
B =
    1.0000 + 1.0000i   2.0000 + 1.0000i
    1.0000 + 2.0000i   2.0000 + 2.0000i
》C = conj(Z')
C =
    1.0000 + 1.0000i   2.0000 + 1.0000i
    1.0000 + 2.0000i   2.0000 + 2.0000i
```

1.6.2 矩阵的加法与减法运算

在 MATLAB 中,矩阵的加法和减法运算分别使用"+"和"-"运算符。要注意的是,只有相同维数的矩阵才能够进行加法、减法运算。例如:

```
》A = [ 11, 22, 33 ; 44, 55, 66 ];
》B = [ 1, 2, 3 ; 4, 5, 6 ];
》C = A + B
C =
    12   24   36
    48   60   72
》D = A - B
D =
    10   20   30
    40   50   60
```

此外,MATLAB 也允许标量与矩阵进行加减法运算。例如:

```
》X = [ 11  21  31 ];
》Y = X - 1
Y =
    10  20  30
```

由上例可见,标量与矩阵之间的加减法运算是这个标量分别与这个矩阵的各个元素进行加法或减法运算。

1.6.3 矩阵的乘法运算

与计算机中的书写形式一致,矩阵乘法用“ * ”表示。要注意的是,只有当两个矩阵中前一矩阵的列数和后一矩阵的行数相同时,才可以进行乘法运算。例如:

```
》A = [ 1  2  3 ; 4  5  6 ];
》B = [ 1  2 ; 1  2 ; 1  2 ];
》C = A * B
C =
    6  12
    15  30
```

另外,两个维数相同的向量的内积(数学上也称为点积或标量积)也可以用这种乘法来实现。例如:

```
》x = [ 1  2  3  4  5 ];
》y = [ 1  3  5  7  9 ];
》s = x * y'
s =
    95
```

在 MATLAB 中,还可以进行矩阵和标量相乘,并且这个标量既可以是乘数,也可以是被乘数。矩阵和标量相乘是将该矩阵中的每一个元素都与该标量相乘。

1.6.4 矩阵的除法运算

在 MATLAB 中,分别用两种矩阵除法运算符号“/”和“\”来表示矩阵的左除和右除运算。只有当 ***A*** 矩阵是非奇异方阵时,A\B 和 B/A 运算才可以实现。

A\B 运算等效于 ***A*** 的逆左乘 ***B*** 矩阵,也就是 inv(A) * B,通常 X=A\B 是 A * X=B 的解。而 B/A 运算等效于 ***A*** 矩阵的逆右乘 ***B*** 矩阵,也就是 B * inv(A),通常 X=A/B 是 X * A=B 的解。

要说明的是,一般情况下,A\B 不等于 A/B。例如:

```
》A = [ 10  20 ;20  10 ];
》B = [ 1 2 ]';
》C = A\B
C =
    0.1000
    0
```

1.6.5 矩阵的乘方

在 MATLAB 中,"A ^ p"表示 A 的 p 次方。如果 A 是一个方阵,p 是一个标量,且 p 是一个大于 1 的整数,则 *A* 的 p 次幂即为 A 自乘 p 次。如果 p 不是整数,则计算将会涉及到特征值和特征向量的问题。例如,若[V,D]=eig(A),则

$$A\hat{}p = V * D.\hat{}p/V$$

其中 V 和 D 分别为矩阵 *A* 的特征向量矩阵和特征值矩阵。例如:

```
》A = [ 10  20 ;20  10 ];
》B = A ^ 3
B =
    13000  14000
    14000  13000
```

另外,如果 p 是矩阵而 A 是标量,或者 A,p 都是矩阵,则 A ^ p 是不能成立的。

1.6.6 MATLAB 定义的点运算

有时,我们需要对两个矩阵各个元素之间进行乘、除、乘方等运算。MATLAB 对这种计算定义了一个新的运算符——点运算符。这个点运算符是在矩阵的乘、除等运算符号之前分别加上". "来表示的,即". x,. \,. /,. ^ "等。

点运算是两个维数相同矩阵对应元素之间的乘、除、乘方等运算,要注意它们与通常矩阵乘、除、乘方运算的不同。例如:

```
》A = [ 10  20  30 ; 40  50  60 ];
》B = [ 2  2  2 ;3  3  3 ];
》C = A. * B
C =
    20  40  60
    120  150  180
》D = A. /B
D =
     5.0000  10.0000  15.0000
    13.3333  16.6667  20.0000
》E = A. ^ B
E =
        100      400      900
      64000   125000   216000
```

1.7 矩阵函数

MATLAB 提供了很多以矩阵为对象的运算函数，丰富了其数学运算能力。

1.7.1 对矩阵的几种基本变换操作

MATLAB 可以通过简单的“ '”运算符实现矩阵的转置运算。例如：

```
》A = [ 11,22,33;44,55,66;77,88,99 ];
》A'
ans =
        11   44   77
        22   55   88
        33   66   99
```

我们可以利用 fliplr( )函数进行矩阵的左右反转。例如：

```
》fliplr(A)
ans =
     33   22   11
     66   55   44
     99   88   77
```

可以利用 flipud( )函数进行矩阵的上下反转。例如：

```
》flipud(A)
ans =
        77   88   99
        44   55   66
        11   22   33
```

可以利用 rot90( )函数将矩阵旋转 90 度。例如：

```
》rot90(A)
ans =
        33   66   99
        22   55   88
        11   44   77
```

我们还可以利用 triu( )函数或 tril( )函数来求一个矩阵的上三角矩阵或下三角矩阵。例如：

```
》triu(A)
ans =
    11   22   33
     0   55   66
     0    0   99
》tril(A)
ans =
    11    0    0
    44   55    0
    77   88   99
```

1.7.2 三角分解

对矩阵所进行的最基本的分解是将 1 个方阵表示成两个基本三角阵的乘积，其中 1 个三角阵为上三角阵，另 1 个为下三角阵。这种分解通常被称为 LU 分解。这里使用的算法是高斯变量消去法，分解的因数可由 MATLAB 的 lu( )函数得到。

- 格式：

  [L , U] = lu(X)

- 功能：

对矩阵 ***X*** 进行 LU 分解。

例如，下面的命令对矩阵 ***A*** 进行 LU 分解，并对分解的结果进行了验证。

```
》A = [1  2  3
       4  5  6
       7  8  0];
》[L,U] = lu(A)
L =
    0.1429   1.0000        0
    0.5714   0.5000   1.0000
    1.0000        0        0
U =
    7.0000   8.0000        0
         0   0.8571   3.0000
         0        0   4.5000
》L * U
ans =
       1  2  3
       4  5  6
       7  8  0
```

上面的例子中，***L*** 是转换了的对角线为 1 的下三角矩阵，***U*** 是上三角矩阵。分解的逆运算

L * ***U*** 证明了该矩阵 LU 分解的正确性。

1.7.3 正交分解

矩阵的正交分解是指将 1 个方阵或长方矩阵表示为 1 个正交矩阵和 1 个上三角矩阵的乘积。正交分解也称为 QR 分解，它可由 MATLAB 的 qr( )函数进行分解。

- 格式：

  [Q , R] = qr(X)

- 功能：

对矩阵 ***X*** 进行 QR 分解。

例如，下面的命令对矩阵 ***A*** 进行 QR 分解。

```
      [ 1   2   3
        4   5   6
》A =
        7   8   9
       10  11  12  ];
》[Q , R] = qr(A)
Q =
    -0.0776   -0.8331   -0.2636   -0.4801
    -0.3105   -0.4512    0.7093    0.4437
    -0.5433   -0.0694   -0.6278    0.5530
    -0.7762    0.3124    0.1821   -0.5166
R =
   -12.8841   -14.5916   -16.2992
          0    -1.0413    -2.0826
          0          0     0.0000
          0          0          0
```

容易验证 Q * R 的值就是原矩阵 ***A***。观察 ***R*** 矩阵的三角结构，其对角线的下半部分全是 0，在对角线上 R(3,3)元素为零值。说明 ***R*** 与原来 ***A*** 矩阵不是满秩的。QR 分解可以解决方程数多于未知数的线性系统问题。

1.7.4 奇异值分解

矩阵的奇异值分解是指将 1 个矩阵分解为 3 个因数矩阵 ***U***、***S*** 和 ***V***，并且使得 A=U * S * V，其中 ***U*** 矩阵和 ***V*** 矩阵是正交矩阵，***S*** 矩阵是对角矩阵。***S*** 矩阵的对角元素，就是 A 的奇异值。

MATLAB 提供了对矩阵进行奇异值分解的函数 svd( )，其用法如下：

- 格式：

  [U , S , V] = svd(X)

- 功能：

对矩阵 ***X*** 进行奇异值分解，返回其奇异值。

例如：

```
》A = [ 1  2  3 ;4  5  6 ;7  8  9 ;10  11  12 ];
》svd(A)
ans =
    25.4624
     1.2907
     0.0000
```

1.7.5　矩阵求逆

在求解线性方程组等运算中,都要用到矩阵求逆的过程。MATLAB 提供了矩阵求逆的函数。

- 格式:

variable = inv(X)

- 功能:

求矩阵 ***X*** 的逆矩阵。

例如:

```
》A = [ 1  2  3
        4  5  6
        7  8  0 ];
》inv(A)
ans =
   -1.7778    0.8889   -0.1111
    1.5556   -0.7778    0.2222
   -0.1111    0.2222   -0.1111
```

1.7.6　求矩阵特征值

如果 ***A*** 是 n×n 矩阵,存在 n 个 λ 值满足式 $Ax=\lambda x$,则 λ 称为矩阵 ***A*** 的特征值,对应的 ***x*** 向量成为 ***A*** 的特征向量。MATLAB 提供了求矩阵特征值及特征向量的函数 eig( )。

计算特征值的函数 eig( )以列向量的形式返回特征值。

- 格式:

variable = eig(A)

[X , D] = eig(A)

- 功能:

求矩阵 ***A*** 的特征值及特征向量。其中,variable 返回的是特征值,X 的行向量为特征向量,D 的对角线数据为特征值。

例如:

```
»A = [ 1  2 ;5  8];
»eig(A)
ans =
   -0.2170
    9.2170
»[ X , D ] = eig(A)
X =
   -0.8543   -0.2365
    0.5198   -0.9716
D =
   -0.2170        0
         0   9.2170
```

1.7.7 矩阵的超越函数

在 MATLAB 中,exp( ),sqrt( )等函数的作用对象是某个指定矩阵的单个元素,即它们分别对矩阵的每 1 个元素进行计算。由这些函数名后面再加上字母 m 就构成了所谓矩阵超越函数,如 expm( ),sqrtm( ),logm( )等,它们分别为矩阵指数函数、矩阵开方函数、矩阵对数函数。矩阵超越函数的计算对象是矩阵,并且它要求这个矩阵必须是方阵。

例如,函数 expm(A)是以矩阵 **A** 为变量来计算下式:

$$\text{expm}(A)=e^{A}=I+\frac{A}{1!}+\frac{A^2}{2!}+\frac{A^3}{3!}+\cdots$$

注意,exp(A)和 expm(A)的运算结果是不同的。例如:

```
»A = [ 1  2 ; 2  1 ];
»exp(A)
ans =
    2.7183   7.3891
    7.3891   2.7183
»expm(A)
ans =
   10.2267    9.8588
    9.8588   10.2267
```

1.8 字符串及其处理

MATLAB 语言也可以处理字符串。在 MATLAB 的工作空间中,字符串是以向量的形式存储的:在 MATLAB 的命令书写中,我们用一对单引号('')所包括起来的内容来表示该字符串。例如,下面的语句中,变量 s 就是 1 个表示字符串的变量。

```
》s = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
s =
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

字符串也可作为矩阵中的元素,此时各个字符串将按下述方式联结成 1 个新的字符串。

```
》s1 = 'STR11' ; s2 = 'STR12';
》s3 = 'STR21' ; s4 = 'STR22';
》s = [ s1 , s2 ;s3 , s4 ]
s =
STR11STR12
STR21STR22
```

MATLAB 提供了一些与字符串操作相关的函数。

一、求字符串长度

- 格式:

  variable = length(str)

- 功能:

返回字符串 str 的长度值到 variable 变量。

例如:

```
》s = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    》m = length(s)
    m =
        26
```

二、字符串与 ASCII 码之间的相互转换

- 格式:

  s = abs(str)

- 功能:

将字符串 str 中的每一个字母转换成相对应的 ASCII 值。

例如:

```
》s = 'MATLAB  5.1';
》ss = abs(s)
ss =
    77  65  84  76  65  66  32  53  46  49
```

- 格式:

  str = setstr(s)

- 功能:

将以 ASCII 方式显示的字符串 s 转换成以字母方式显示的字符串。

例如:

```
》s = setstr(ss)
s =
MATLAB  5.1
```

三、字符串与数值之间的相互转换

• 格式：

s = num2str(x)

• 功能：

将数值 x 转换为字符串。

例如：

```
》s = pi;
》isstr(s)
ans =
      0
》s = num2str(s);
》isstr(s)
ans =
      1
```

• 格式：

x = str2num(str)

• 功能：

将由数字组成的字符串 str 转换为对应的数值 x。

例如：

```
》x = str2num('12345.12345')
x =
    1.2345e+004
```

• 格式：

s = int2str(n)

• 功能：

将整数 n 转换为字符串。

例如：

```
》s = int2str(999)
s =
999
```

四、字符串比较函数

(1) 比较字符。

• 格式：

ret = strcmp(s1 , s2)

• 功能：

比较字符串 s1 和 s2 是否相同。若相同，则返回 1；若不相同，则返回 0。
例如：

```
》strcmp( 'MATLAB 4.1' , 'MATLAB 5.1' )
ans =
       0
》strcmp( 'MATLAB 5.1' , 'MATLAB 5.1' )
ans =
       1
```

(2) 检查字母。

- 格式：

  ret = isletter(str)

- 功能：

检查字符串 str 中的各个元素是否为字母。若是字母，则返回 1；若不是字母，则返回 0。
例如：

```
》isletter( 'ABCDE12345' )
ans =
       1  1  1  1  1  0  0  0  0  0
```

(3) 检查字符串。

- 格式：

  ret = isstr(str)

- 功能：

检查 str 是否为 1 个字符串。若是 1 个字符串，则返回 1；若不是，则返回 0。
例如：

```
》isstr( 'ABCDE' )
ans =
       1
》isstr(12345)
ans =
       0
```

五、字符串查找与替换函数

(1) 查找。

- 格式：

  index = findstr(s1,s2)

- 功能：

返回字符串 s2 在字符串 s1 中的位置。

例如：

```
》str = 'He is a student and she is a student too.';
》findstr(str , 'He')
ans =
        1
》findstr(str , 'student')
ans =
        9   30
》findstr(str , 'you')
ans =
        [ ]
```

(2) 替换。

· 格式：

s = strrep(s1,s2,s3)

· 功能：

将字符串 s1 中的所有字符串 s2 用字符串 s3 来代替，并返回这个新的字符串。

例如：

```
》str = 'He is a student and she is a student too.';
》strrep( str , 'student' , 'teacher' )
ans =
He is a teacher and she is a teacher too.
```

六、字符串处理函数

(1) 处理成空格。

· 格式：

str = blanks(n)

· 功能：

返回含有 n 个空格的字符串。

例如：

```
》s = [ 'ABCDE' , blanks(3) , 'ABCDE' ]
s =
ABCDE   ABCDE
```

(2) 处理成命令。

· 格式：

eval(s)

· 功能：

将字符串 s 理解为 MATLAB 的 1 个命令来执行。

例如，下面的命令将把工作空间中的全部变量分别保存到文件 Data1. MAT，Data2. MAT、……，Data8. MAT 中。

```
》for n = 1 : 8
    str = [ 'save Data' , int2str(n) ]
    eval(str)
end
```

(3) 大小写转换。

- 格式：

  upper(s)

  lower(s)

- 功能：

upper( )函数将字符串中的小写字母转变为大写字母；lower( )函数将字符串中的大写字母转变为小写字母。

例如：

```
》str = upper( 'AaBbCcDdEe' )
str =
AABBCCDDEE
》str = lower( 'AaBbCcDdEe' )
str =
aabbccddee
```

1.9　MATLAB 的数值分析计算

在科学工作中，我们会经常须要解析计算函数的最大值、最小值、零点等，须要计算函数的积分、微分，也须要绘制其示意图。MATLAB 提供了一些进行这些数值分析计算的函数。

1.9.1　函数图形绘制

当需要对 1 个函数进行分析时，有必要首先绘制出函数的图形。我们可以利用 fplot( )函数来绘制函数图形。

- 格式：

  fplot(fun , range)

- 功能：

绘制出由字符串 fun 所表示函数的图形，绘图范围由 range 向量来说明。

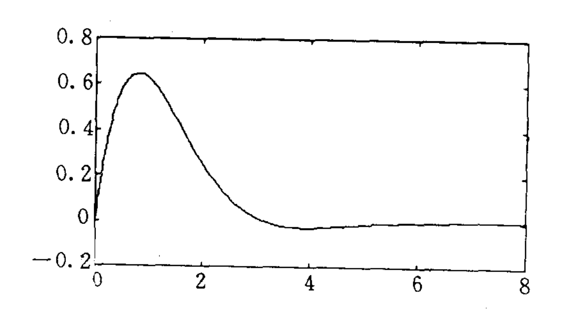


图 1 - 3　函数图形

例如，下面的命令将绘制出图 1 - 3 所示的函数图形。

```
》f = '2 * exp(-x). * sin(x)';
    》fplot(f , [0,8])
```

1.9.2 数值分析计算

· 格式:

y = fmin(fun , x1 , x2)

y = fzero(fun , x)

z = trapz(x , y)

· 功能:

fmin( )函数可求出由字符串 fun 所表示函数在[x1 , x2]范围内的最小点(即函数 fun 取最小值时的 x 值)。求函数最大点(即函数 fun 取最大值时的 x 值)时,须在函数前加上负号,即用['-',fun]代替 fun。

fzero( )函数可求出由字符串 fun 所表示函数在 x 点附近的零点。

trapz( )函数可求出 y 对 x 的积分值。

例如:

```
》f = '2 * exp(-x). * sin(x)';
》xfmin = fmin(f , 0 , 8)
xfmin =
        3.9270
   》xfmax = fmin(['-',f] , 0 , 8)
   xfmax =
        0.7854
   》xfzero = fzero(f , 3)
   xfzero =
        3.1416
        》x = 0 : 0.5 : 8;
        》y = eval(f);
   》z = trapz(x,y)
   z =
        0.9584
```

1.10 MATLAB 的控制语句

与其他计算机高级语言相似,MATLAB 语言也提供了用于控制程序执行顺序的循环语句、条件转移语句,它们使得我们可以简单地实现复杂的操作和运算。

1.10.1 循环语句

MATLAB 提供了两种控制程序循环操作的语句:for 循环和 while 循环。

一、for 循环

for 循环语句可以使一条或一组语句能够以指定的次数反复执行多次。

· 格式:

```
for  variable = StartVal : StepVal : EndVal
     statements
end
```

· 功能：

以 StartVal 为循环控制变量 variable 的初始值，EndVal 为终止值，StepVal 为循环步长，反复执行 for - end 之间的 MATLAB 语句组 statements。

· 注意：

① 当循环步长值 StepVal 为 1 时，可省略不写该步长值。

② for 语句可以嵌套使用。

③ break 命令可终止本级 for - end 循环过程。

例如，可用下述命令求 50×50 的矩阵 ***A*** 的所有元素之和。

```
》sum = 0;
》for i = 1 : 50
    for j = 1 : 50
      sum = sum + A(i,j)
    end
  end
```

再例如，用下述命令可将向量 ***B*** 的所有元素全部赋值为 10^6。

```
》for i = 1 : 500
    B(i) = 10^6
 end
```

这里要说明的是，在 MATLAB 语言中，for 循环的执行速度较慢。而一些 for 循环语句结构可由冒号操作符来实现，冒号操作符的运算速度却是比较快的。例如，上面这个例子的效果与下述命令完全相同。

```
》i = 1 : 500;
》B(i) = 10^6
```

二、while 循环

while 循环可以使一条或一组语句能够在一定条件下反复执行多次。

· 格式：

```
while expression
  statements
end
```

· 功能：

当条件表达式 expression 为真时（逻辑运算结果为 1 时），反复执行 while - end 之间的 MATLAB 语句组 statements。

· 注意：

① while 语句可以嵌套使用，但不能交叉使用。

② break 命令可终止本级 while - end 循环过程。

例如，用下述命令可以计算出 $\sum_{i=1}^{100} i^2$ 的值。

```
»sum = 0 ;
»i = 1;
»while i<=100
    sum = sum + i*i
    i = i + 1
  end
```

1.10.2 条件转移语句

MATLAB 提供了 3 种条件转移语句:简单条件转移语句、复杂条件转移语句和开关条件转移语句。

一、简单条件转移语句

- 格式:

```
if  expression
    statements
end
```

- 功能:

当条件表达式 expression 为真时(逻辑运算结果为 1 时),执行 if - end 之间的 MATLAB 语句组 statements。

例如,下面的命令将计算出向量 ***A*** 中的所有正数之和。

```
»A = [1, 2, 3, -1, -2, -3, 22, 88, 99, …, 888 ];
»sum = 0;
»for i = 1:length(A)
    if  A(i) > 0
        sum = sum + A(i)
    end
  end
```

二、复杂条件转移语句

- 格式:

```
if  expression1
    statements1
elseif  expression2
    statements2
elseif ……
    …………
else
    statements0
end
```

- 功能:

当条件表达式 expression1 为真时,执行语句组 statements1;否则,当条件表达式 expression2 为真时,执行语句组 statements2……否则,当所有的条件表达式都不为真时,执行语句组 statements0。

例如，用下面的命令计算 $y=\begin{cases}100x-x^2, & x>0 \\ 0, & x=0 \\ x^3, & x<0\end{cases}$ 的值。

```
》if x > 0
    y = 100 * x－x * x
  elseif x == 0
    y = 0
  else
    y = x * x * x
  end
```

三、开关条件转移语句

• 格式：

```
switch expression
  case value1
    statements1
  case value2
    statements2
  case ……
    …………
  otherwise
    statements0
end
```

• 功能：

当表达式 expression 的取值为 value1 时，执行语句组 statements1；否则，当表达式 expression 的取值为 value2 时，执行语句组 statements2……否则，执行语句组 statements0。

例如，用下面的命令根据数字来显示文字形式的星期几。

```
NumDate = input('Please Enter a Number(1,2,3,4,5,6,7): ')
switch NumDate
  case 1
    disp( 'Monday' );
  case 2
    disp( 'Tuesday' );
  case 3
    disp( 'Wednesday' );
  case 4
    disp( 'Thursday' );
  case 5
    disp( 'Friday' );
  case 6
    disp( 'Saturday' );
  case 7
    disp( 'Sunday' );
  otherwise
    disp( 'Enter Error' );
end
```

1.11 辅助语句

1.11.1 注释语句

- 格式：

  % text

- 功能：

说明 text 所表示的文字内容是程序的注释内容。

在程序中使用注释语句，可提高程序的可读性。

1.11.2 中断语句

- 格式：

  break

- 功能：

终止 1 个循环语句的执行过程。

break 命令可强制将控制从正在执行的 for 循环或 while 循环中跳出，转去执行循环语句后面的语句。例如，下面的命令直到用户输入了字母 y 或 n 之后才结束循环过程。

```
while(1)
    str = input('Please Enter Your Answer [y/n] : ' , 's');
    if (str=='Y' | str=='y' | str=='N' | str=='n' )
      break;
    end
end
```

1.11.3 暂停语句

- 格式：

  pause

  pause(n)

- 功能：

pause 语句暂时停止程序的执行，直到用户按任意键之后，才继续执行程序；而 pause(n) 语句将暂时停止程序的执行 n 秒钟，n 秒钟之后，程序自动继续执行。

1.11.4 回显控制语句

- 格式：

  echo on/off

- 功能：

控制是否在屏幕上回显 MATLAB 正在执行的语句。

- 注意：

系统所默认的状态是不回显，即 echo off。

echo on 的引入，对我们调试程序有很大的帮助，因为此时我们通过屏幕就可以知道程序当前已运行到哪一步，可以很方便地知道哪一步有问题。但对已调试好的程序，我们一般希望采用 echo off 状态。

1.12 MATLAB 的输入输出语句

1.12.1 输入语句

一、输入数值

- 格式：

  x = input('string')

- 功能：

先将字符串 string 显示在屏幕上，然后等待用户从键盘上输入一个数值到数值型变量 x 中。

例如：

```
》YourAge = input('Please Input Your Age: ')
Please Input Your Age: 18
YourAge =
     18
```

二、输入字符串

- 格式：

  x = input('string' , 's')

- 功能：

先将字符串 string 显示在屏幕上，然后等待用户从键盘上输入一个字符串到字符串型变量 x 中。

例如：

```
》YourName = input('Please Input Your Name: ' , 's')
Please Input Your Name: Jim Zhou
YourName =
Jim Zhou
```

1.12.2 输出语句

一、输出显示命令

- 格式：

  disp(variable)

- 功能：

在屏幕上显示变量 variable 所指定的数值、字符串或矩阵内容。

例如：

```
》disp('My name is Jim Zhou.')
My name is Jim Zhou.
》disp(1+2+3+4+5+6+7+8+9+10)
    55
》A = [ 1, 2, 3 ; 4, 5, 6 ];
》disp(A)
    1   2   3
    4   5   6
```

二、错误信息显示命令

· 格式：

error('string')

· 功能：

在屏幕上显示 string 所表示的错误信息。

例如，下面的语句用于检查除数是否为零。

```
if (x~= 0)
    y=1000/x
else
    error('Overflow Error')
end
```

1.13 MATLAB 的文件操作

1.13.1 变量的保存与调用

一、保留变量到文件

我们可以用 save 命令将 MATLAB 工作空间中的变量保存到磁盘文件中，以便于以后调用这些变量。

· 格式：

save filename variables

· 功能：

将变量列表 variables 所列出的变量保存到磁盘文件 filename 中。

· 说明：

① variables 所表示的变量列表中，不能使用逗号，各个不同的变量之间只能用空格来分隔。

② 未列出 variables 时，表示将当前工作空间中的所有变量都保存到磁盘文件中。

③ 指定磁盘文件的扩展名为“.mat”。一般，这是 1 个二进制的文件。

例如，下面的命令将保存 x,y,z 三个变量。

```
》save   fname   x   y   z
```

二、从文件中读取变量

我们可以用 load 命令将已经保存在磁盘文件中的变量再调回到 MATLAB 的工作空间中。这种方法可以节省工作空间。

· 格式:

load filename variables

· 功能:

将以前用 save 命令保存的变量 variables 再从磁盘文件调入 MATLAB 的工作空间。

· 说明:

① 用 load 命令调入的变量,其名称为用 save 命令保存时的名称,取值也一样。

② variables 所表示的变量列表中,不能使用逗号,各个不同的变量之间只能用空格来分隔。当未列出 variables 时,表示将磁盘文件中的所有变量都调入工作空间。

例如,下面的命令将调入文件"fname. mat"中所保存的所有变量。

```
》load fname
```

1.13.2 文件的打开与关闭

前面介绍的 save 和 load 命令是处理扩展名为". mat"的文件。MATLAB 也提供了另外的文件打开、关闭和处理命令,它能够使我们对普通的文件进行操作。

一、打开一个文件

· 格式:

fp = fopen(fname , ftype)

· 功能:

打开一个指定的文件,并返回其句柄值。

· 注意:

① fp 为文件能够正常打开后所返回的句柄值。文件句柄可以理解为文件的 1 个临时标识符,以后就可利用它来指明对哪个文件进行处理。如果打开文件的操作失败,则返回的句柄值为-1。

② fname 是 1 个表示文件名称的字符串。

③ ftype 是 1 个表示文件类型的字符串,如'r'表示只读型的文件,'w'表示只写型的文件,而'a'表示可添加型的文件。

例如,下面的命令分别打开了可用于读和写的文件。

```
》fp1 = fopen( 'DATA1. TXT' , 'r' );
》fp2 = fopen( 'DATA2. TXT' , 'w' );
```

二、关闭一个文件

· 格式:

st = fclose(fp)

· 功能:

关闭文件句柄 fp 所指定的文件。

· 注意：

如果该文件不存在，则返回值为－1，但这种情况下并不会中断程序的运行过程。

1.13.3 文件的输入与输出

一、从文件读出数据

(1) 不变格式读取。

· 格式：

A = fread(fp , size)

· 功能：

从文件句柄 fp 所指定的文件中读取数据，并将这些数据保存到矩阵 **A** 中。

· 注意：

① 参数 fp 是由 fopen 命令所打开文件的句柄。

② 参数 size 指定了从文件中读取多少个数据。若不指定该参数，则读取全部文件的数据。也可只读取其中的几个数据。

例如：

```
》fp = fopen( 'fread.m' , 'r' )
》F = fread( fp )
》s = setstr( F' )
```

(2) 指定格式读取。

· 格式：

A = fscanf(fp , format , size)

· 功能：

从文件句柄 fp 所指定的文件中，按照字符串 format 所指定的数据格式读取数据，并将这些数据保存到矩阵 **A** 中。

· 注意：

① 参数 fp 是由 fopen 命令所打开文件的句柄。

② 字符串参数 format 指定了数据的读取格式，该格式的书写方式及含义与 C 语言中的格式说明方式相同，即在“%”号后再跟上 d，i，o，u，x，e，f，g，s，c 等修饰字母。

③ 参数 size 指定了最多能从文件中读取多少个数据。若不指定该参数，则读取全部文件的数据。也可只读取其中的部分数据。

例如：

```
》Str = fscanf(fp , '%s')                 % 读取一个字符串
》IntNum = fscanf(fp , '%5d')             % 读取一个整数
》FloatNum = fscanf(fp , '%5.2f')         % 读取一个浮点数
```

二、将数据写入文件

· 格式：

fprintf(fp , format , A , B , …)

· 功能：

按照字符串 format 所指定的数据格式，依次将数据 A，B，…写入到由文件句柄 fp 所指定的文件中。

· 注意：

① 参数 fp 是由 fopen 命令所打开文件的句柄。也可指定 fp 等于 1，此时是指向屏幕输出数据。

② 字符串参数 format 指定了数据的输出格式，该格式的书写方式及含义与 C 语言中的格式说明方式相同，即在“%”号后再跟上 d，i，o，u，x，e，f，g，s，c 等修饰字母。

③ 在 format 参数中，也可使用 '\n' 这个控制符，以便输出一个回车换行符。

例如，下面的命令将会生成一个下述排列形式的文件。

```
》x = 0 ; 0.1 ; 1;
》y = [ x ; exp(x) ];
》fp = fopen('FunExp.TXT' , 'w');
》fprintf(fp , 'x=%6.2f      exp(x)=%12.8f\n' , y);
》fclose(fp);
```

```
x= 0.00    exp(x)= 1.00000000
x= 0.10    exp(x)= 1.10517092
x= 0.20    exp(x)= 1.22140276
x= 0.30    exp(x)= 1.34985881
x= 0.40    exp(x)= 1.49182470
x= 0.50    exp(x)= 1.64872127
x= 0.60    exp(x)= 1.82211880
x= 0.70    exp(x)= 2.01375271
x= 0.80    exp(x)= 2.22554093
x= 0.90    exp(x)= 2.45960311
x= 1.00    exp(x)= 2.71828183
```

FunExp.TXT 文件中的内容

1.14 M 程序与 M 函数

1.14.1 MATLAB 程序

在 MATLAB 中，我们可以把它的一些命令按工作要求组合在一起，这些命令按一定的顺序执行，这就构成了所谓的程序。MATLAB 的程序都是以扩展名为“.m”的形式存放在磁盘中的，但细分起来，这些 MATLAB 程序又有两种不同的类型。

一种是由保留关键字“function”为开头所形成的 MATLAB 程序，它完成某一特定的功能。这种程序称为“函数(function)”，简称为 M 函数。

另一种是普通的 MATLAB 程序，它是由 MATLAB 的一些命令所组合而成的，其首行不是由 MATLAB 的保留关键字“function”所开头，而是普通的 MATLAN 命令。这种程序称为“正本文档(script)”，简称为 M 程序。这类 M 程序在 MATLAB 提示符后输入其文件名，即可执行该程序，它一般可作为系统的主程序。

例如，下面所示为1个简单的求长方形面积的MATLAB程序。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX101.M
% 功能:计算长方形的面积
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Length = input('请输入长方形的长度:');
Width = input('请输入长方形的宽度:');
disp(['长方形的面积 = ',int2str(Length * Width)]);
```

我们在MATLAB命令窗口中键入这个程序的名称即可执行该程序。

```
》EX101
请输入长方形的长度:123
请输入长方形的宽度:789
长方形的面积 = 97047
```

1.14.2 MATLAB函数

引入M函数的优点是提高程序的可读性及移植性，它类似于其他高级计算机语言中的子程序或函数。

M函数的基本格式:

- 格式:

```
function variables = fname(input - variables)
% text              注释说明语句
statement1
statement2          函数体语句
…………
```

- 功能:

定义1个M函数。

- 说明:

① M函数的输入变量个数被记忆在MATLAB的保留参数nargin中；M函数的输出变量个数被记忆在MATLAB的保留参数nargout中。

② 在函数体语句中，必须要对各个返回变量至少赋值一次。

③ 返回变量个数如果多于1个时，则必须用方括号把它们括起来。

④ M函数是以fname为文件主名存放在磁盘中的。

⑤ 当程序执行到M函数的末尾时，将返回到调用处。也可使用return语句来强行从被调用的M函数中返回。

例如，下面所示程序是求矩阵元素平方和的函数。

```
function y = SquSum(x)
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:SquSum.M
% 功能:求矩阵元素的平方和
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
[m,n] = size(x);
y = 0;
for i = 1:m
  for j = 1:n
    y = y+x(i,j) * x(i,j);
  end
end
```

引用函数的名称并代入相应的变量，即可调用 M 函数。例如，上述 M 程序的调用方式如下所示：

```
》A = [ 1 , 2 , 3 , 4 , 5 ];
》B = [ 1 , 1 , 1 ; 6 , 6 , 6 ; 8 , 8 , 8 ];
》Asum = SquSum(A)
Asum =
    55
》Bsum = SquSum(B)
Bsum =
    303
```

1.14.3 全局变量与局部变量

在 M 程序和 M 函数中，我们都会使用一些变量，但 M 程序中的变量和 M 函数中的变量却有着很大的区别：M 程序中的所有变量是全局起作用的（全局变量），而 M 函数中的变量却是局部起作用的（局部变量）。在函数返回后，这些局部变量会自动在 MATLAB 的工作空间中清除掉。

但假如函数内部的某些变量需要在函数外使用，则我们应通过命令将它们设置为全局变量。全局变量是由 global 命令来设置的。

- 格式：

  global variables

- 功能：

指定变量列表 variables 所列出的全部变量为全局变量。

例如，在下面的示例程序中，str1 为全局变量，而 str2 是局部变量。

```
function GlobStr
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：GlobStr.M
% 功能：全局变量与局部变量示意
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
global str1
str1='I am a global variable'
str2='I am not a global variable'
```

在 MATLAB 命令窗口中键入下述命令，可检验出 str2 未定义，它是一个只在 GlobStr.M 函数中起作用的局部变量。

```
》global str1
》GlobStr                    % 调用 GlobStr 函数
str1 =
I am a global variable
str2 =
I am not a global variable
》str1                       % 检查是否已经定义了变量 str1
str1 =
I am a global variable
》str2                       % 检查是否已经定义了变量 str2
??? Undefined function or variable 'str2'.
```

第2章　二维图形绘制

在科学研究及日常工作中，将一些数据资料用图形的方式显示出来，将会使人们更直观、更形象地了解问题。以前，我们用高级语言绘制图形时，通常的方法是先对数据进行预处理，找出其最大值、最小值，然后再根据需要确定各个坐标轴的范围、刻度大小等，最后才能用系统所提供的一些低级绘图命令在屏幕上绘制出图形。这种绘图方法不仅比较麻烦，更重要的是，它缺乏通用性，我们很难简单地找到一个较好的既面面俱到又适用于很大数据范围的通用绘图方法。

但MATLAB的出现，却使得我们可以既简单又方便地进行各种图形的绘制工作。MATLAB所提供的丰富的绘图功能，使得我们从烦琐的绘图细节中脱离出来，而能够专心于最关心的问题。本章主要介绍MATLAB所提供的一些二维绘图命令及其使用方法，具体包括以下内容：

- 基本二维绘图命令
- 二维图形的修饰方法
- 函数图形的精确绘制
- 图形的填充方法
- 多坐标系绘图与图形窗口的分割方法
- 特殊坐标图形的绘图方法
- 特殊二维图形的绘制方法

2.1　二维图形的基本绘图命令

MATLAB提供了两种级别的二维图形基本绘图命令。一种是高级绘图命令plot，它可以使我们能够以一体化的方式绘出图形，即用户只须给出图形定义数据，绘图范围、刻度大小等，细节内容都可由系统自动确定；另一种是低级绘图命令line，它可使我们能够单独地绘制图形的细节内容。

2.1.1 高级绘图命令

- 格式：

  plot(x1 , y1 , option1 , x2 , y2 , option2 ,…)
- 功能：

x1 ，y1 所给出的数据分别为 x ，y 坐标值，option1 为选项参数，以逐点连折线的方式绘制 1 个二维图形；同时，x2 ，y2 所给出的数据分别为 x ，y 坐标值，option2 为选项参数，以逐

点连折线的方式绘制另 1 个二维图形……

· 说明：

plot 命令是将各个数据点通过连折线的方式来绘制二维图形的，画直线是其基本绘图步骤。其实，对于曲线，若把它进行细分的话，也可看作是由直线相连接而构成的。

上面给出的 plot 命令格式是一种完整的格式，在实际操作中，根据各个 x 数据的取法，以及 x ，y 是向量还是矩阵等情况，均可以有较简单的书写格式。现分别叙述如下：

(1) x 、y 数据均指定时，plot 命令的格式为：

plot(x ，y)

这种书写格式的具体执行情况可分为下列几种：

① x ，y 都是向量时，则本命令绘制出表示 *x* ，*y* 之间关系的二维折线图形。

② 当 x 表示 1 个向量，而 y 表示 1 个矩阵时，若 *y* 矩阵的行向量长度与 *x* 向量的长度一致时，则本命令绘制出 *y* 矩阵的各个行向量相对于 *x* 的 1 组二维折线图形；若 *y* 矩阵的列向量长度与向量 *x* 的长度一致时，则本命令绘制出 *y* 矩阵的各个列向量相对于 *x* 向量的 1 组二维折线图形。

③ 当 x 表示 1 个矩阵，而 y 表示 1 个向量时，若 *x* 矩阵的行向量长度与 *y* 向量的长度一致时，则本命令绘制出 *y* 向量相对于 *x* 矩阵的各个行向量的 1 组二维折线图形；若 *x* 矩阵的列向量长度与向量 *y* 的长度一致时，则本命令绘制出 *y* 向量相对于 *x* 矩阵的各个列向量的 1 组二维折线图形。

④ 当 x ，y 都是维数相同的矩阵值时，则本命令绘制出 *y* 矩阵的每 1 个行向量与 *x* 矩阵的每 1 个行向量之间关系的 1 组二维图形。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2－1 所示图形。

```
》x = 0 : pi/100 : 2 * pi;
》y1 = sin(x);
》y2 = cos(x);
》plot( x , y1 , x , y2 )
```

(2) 不指定 x 数据，而只指定 y 数据时，plot 命令的格式为：

plot(y)

这种书写格式的具体执行情况可分为下列几种：

① 若 y 为向量值时，则以 x=1,2,3,…为各个数据点的 x 坐标，以 *y* 向量的各个对应元素为 y 坐标，画出 1 个二维折线图形。

② 若 y 表示 1 个实数矩阵时，则以 x=1,2,3,…为各个数据点的 x 坐标，分别以 *y* 矩阵的各个行向量为对应的 y 坐标轴值，画出 1 组二维折线图形。

③ 若 y 为复数值时，则以其实部为 X 轴，虚部为 Y 轴，绘出 1 个二维折线图形，即此时相当于执行命令 plot(real(y) , imag(y))。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2－2 所示的锯齿图形。

```
》y = [ 0 , 2 , 0 , 2 , 0 , 2 , 0 , 2 , 0 ];
》plot(y)
```

(3) 绘图命令中还含有一些选项参数时，plot 命令的格式为：

plot(x , y , option)

这种书写格式的绘图命令基本上与 plot(x , y)形式的绘图命令的执行情况相类似，它们的区别在于前者的选项参数 option 指明了所绘图形中线条的线型、颜色以及各个数据点的表示记号。这样，当多条曲线绘制在同一个图形窗口中时，可以用不同的线型和颜色将它们方便地区分开来。

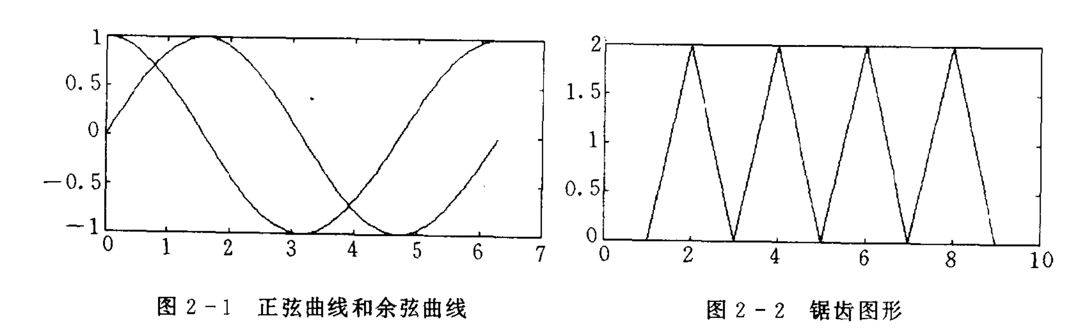


图 2－1　正弦曲线和余弦曲线　　　　图 2－2　锯齿图形

选项参数 option 是由一对单引号所包括起来的 1 个或几个符号或字符表示的，这些字符的具体含义如表 2－1 所示。MATLAB 的缺省设置是黄色的实线，即′－y′。

表 2－1　线型、数据点标记及颜色选项的含义

线型、数据点标记	含　义	颜　色	含　义
.	用点号绘制各个数据点	y	黄色
。	用圆圈绘制各个数据点	m	洋红色
x	用叉号绘制各个数据点	c	蓝绿色
+	用加号绘制各个数据点	r	红色
*	用星号绘制各个数据点	g	绿色
—	实线	b	蓝色
:	点线	w	白色
—.	点划线	k	黑色
— —	虚线		

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2－3 所示的不同线型、不同数据点记号的图形。

```
》x = 0 : pi/20 : 2 * pi;
》y1 = sin(x);
》y2 = sin(x+pi/2);
》plot( x , y1 , 'r:' , x , y2 , '+' )
```

2.1.2　低级绘图命令

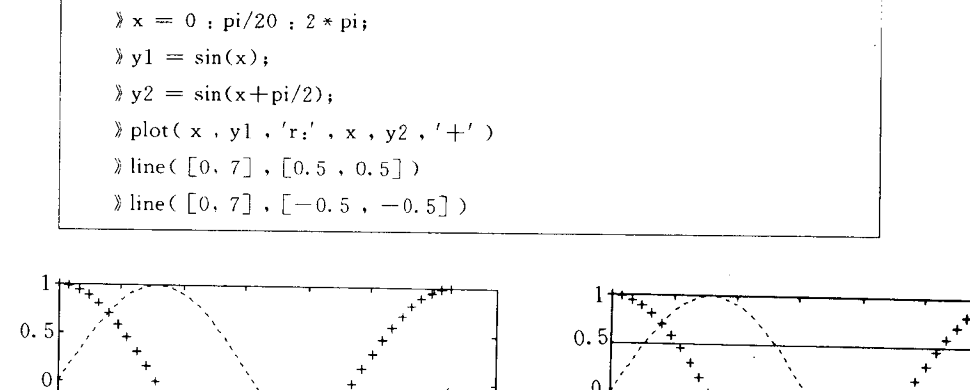
MATLAB 允许用户在图形窗口的任意位置用低级绘图命令 line 画直线或折线。

· 格式：

line(x , y)

· 功能：

在当前图形窗口中绘制出一条由向量 *x* 和向量 *y* 的对应数据元素为数据点的折线。例如，下述命令可在图 2－3 中加上两条水平直线，结果如图 2－4 所示。



```
》x = 0 : pi/20 : 2 * pi;
》y1 = sin(x);
》y2 = sin(x+pi/2);
》plot( x , y1 , 'r:' , x , y2 , '+' )
》line( [0, 7] , [0.5 , 0.5] )
》line( [0, 7] , [-0.5 , -0.5] )
```

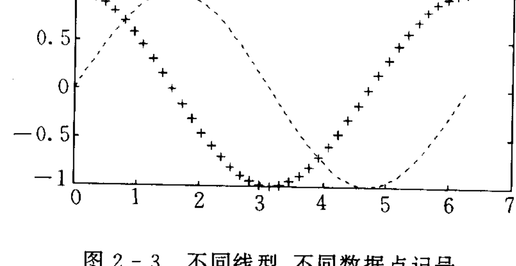


图 2－3　不同线型、不同数据点记号、不同颜色的曲线

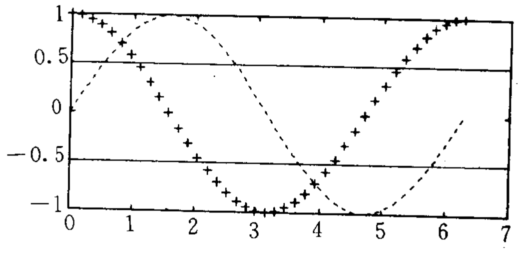


图 2－4　不同线型、不同数据点记号、不同颜色的曲线

2.2　二维图形的修饰

由前面的图 2－1至图 2－4 可以看出，当利用 plot 命令绘图时，虽然运用起来比较简单，但它所自动产生出的图形却显得有些简单，未能产生特殊的效果。为此，MATLAB 提供了一些图形函数，专门用于对由 plot 命令所画出的图形进行进一步修饰，以使其更加美观、更便于应用。如坐标轴范围的设定(axis 命令)、加坐标轴名称(xlabel ，ylabel 命令)、加网格(grid 命令)、给图形加图题(title 命令)、对图形进行文字注释(text 命令)等。下面分别进行说明。

2.2.1　坐标轴的调整

MATLAB 可以自动根据曲线数据的范围选择合适的坐标系，从而使得曲线能够尽可能清晰地显示出来，所以在一般情况下用户不必去进行坐标系的选择。但是，如果用户对 MATLAB 所自动产生的坐标轴不太满意的话，则可以利用 axis 命令来对所要绘制出的图形的坐标轴进行调整。

axis 命令的功能非常丰富，按常用用法分有如下 4 类：调整坐标轴的范围、调整坐标轴的状态、保存调整坐标轴的范围、保存坐标轴的状态。

一、调整坐标轴的范围

- 格式：

  axis([xmin xmax ymin ymax])

- 功能：

将所画图形的 X 轴的大小范围限定在{xmin ，xmax}之间，Y 轴的大小范围限定在

{ymin，ymax}之间。

· 说明：

在绘图时，由于图形的坐标已经给定，所以对坐标轴范围参数的更改，其实际效果也就相当于对原图形进行了放大或缩小处理。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2-5 所示图形。

```
》x = 0 : pi/100 : 2*pi;
》y = sin(x);
》line( [ 0 , 2*pi ] , [0 , 0] )
》plot( x , y )
》axis( [ 0  2*pi  -1  1 ] )
```

而如果我们将最后一条命令改为：

axis([0 2*pi -2 2])

就可画出如图 2-6 所示的图形，其显示效果就好象对图 2-5 的 Y 向进行了压缩。

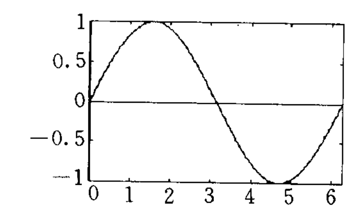


图 2-5 坐标轴调整前的图形

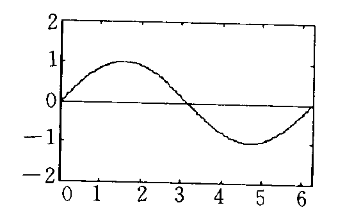


图 2-6 坐标轴调整后的图形

二、调整坐标轴的状态

· 格式：

axis(str)

· 功能：

将坐标轴的状态设定为字符串参数 str 所指定的状态。

· 说明：

① 参数 str 是由一对单引号(' ')所包括起来的字符串(也可省略这对单引号)，它表明了将坐标轴调整为哪一种状态。各种常用字符串的含义如表 2-2 所示。

表 2-2 函数 axis( )的常用功能

命令形式	命令功能
axis([xmin xmax ymin ymax])	按照用户给出的 X 轴和 Y 轴的最大、最小值选择坐标系
axis auto	自动设置坐标系：xmin=min(x)； xmax=max(x)； ymin=min(y)； ymax=max(y)
axis('auto')	自动设置坐标系：xmin=min(x)； xmax=max(x)； ymin=min(y)； ymax=max(y)

续 表

命令形式	命　令　功　能
axis(axis)	将当前的坐标范围值固定下来，使得 hold 开关打开，在这之后的图形都采用当前坐标范围值
axis xy	使用笛卡尔坐标系
axis('xy')	使用笛卡尔坐标系
axis ij	使用 matirx 坐标系。即：坐标原点在左上方，x 坐标从左向右增大，y 坐标从上向下增大
axis('ij')	使用 matrix 坐标系。即：坐标原点在左上方，x 坐标从左向右增大，y 纵坐标从上向下增大
axis square	将当前图形设置为正方形图形
axis('square')	将当前图形设置为正方形图形
axis equal	将 x，y 坐标轴的单位刻度设置为相等
axis('equal')	将 x，y 坐标轴的单位刻度设置为相等
axis normal	关闭 axis equal 和 axis square 命令的作用
axis('normal')	关闭 axis equal 和 axis square 命令的作用
axis off	关闭网格线、xy 坐标的用 label 命令所加的注释，但保留用图形中用 text 命令和 gtext 命令所添加的文本说明
axis('off')	关闭网格线、xy 坐标的用 label 命令所加的注释，但保留用图形中用 text 命令和 gtext 命令所添加的文本说明
axis on	打开网格线、xy 坐标的用 label 命令所加的注释
axis('on')	打开网格线、xy 坐标的用 label 命令所加的注释

（2）只要不产生矛盾的含义，一些不同的参数 str 可以同时起作用，即我们在语句 axis(str)的参数中，可以使用多个修饰字符串，如 axis('auto' ，'on' ，'ij')。

例如，图 2－7 所示正弦曲线图形使用的是笛卡儿坐标系，它可由下述命令来设置：

axis('xy')　或　axis xy

笛卡儿坐标系的图形，其坐标系的 X 轴是由左向右，Y 轴是由下向上的。而有时我们需要绘制 Y 轴是由上向下的图形，即符合屏幕坐标系的图形，这时可由下述命令来设置坐标轴的方向：

axis('ij')　或　axis ij

此时所绘制出的正弦曲线图形如图 2－8 所示。

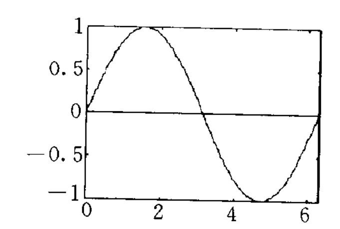


图 2－7　笛卡儿坐标系下的正弦曲线

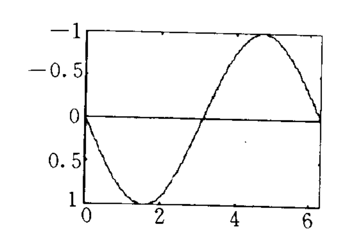


图 2－8　屏幕坐标系下的正弦曲线

又例如，下述程序将绘制出一个单位圆，如图 2 - 9 所示。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX201.M
% 功能:绘制单位圆
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
alpha = 0:0.01:2*pi;
x = sin(alpha);
y = cos(alpha);
plot(x,y)
axis( [ -1.5  1.5  -1.5  1.5 ] )
grid on
```

仔细观察图 2 - 9 可知，这个单位圆有点像一个椭圆，这主要是由于计算机屏幕上 X 方向和 Y 方向的单位长度不一致所造成的。但下述命令却可以消除这种不一致，从而可以绘制出一个真正的单位圆(如图 2 - 10 所示)。

axis('square')　或　axis square

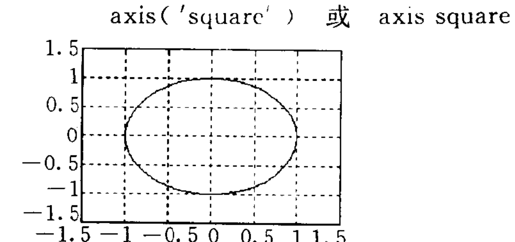


图 2 - 9　未进行刻度调整的单位圆

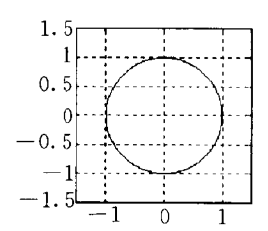


图 2 - 10 用 axis('square')调整刻度后的单位圆

下述命令也可以消除这种不一致，从而可以绘制出一个标准的单位圆(如图 2 - 11 所示)。

axis('equal')　或　axis equal

注意，axis('square')的含义是将 X 坐标轴长度与 Y 坐标轴长度调整为正方形，而 axis('equal')的含义是将 X 坐标轴和 Y 坐标轴的单位刻度大小调整一样长短。

若用下述命令却可以将图形恢复显示为刻度调整前的形式(图 2 - 9)。

axis('normal')　或　axis normal

用下述命令可以关闭或打开图形的坐标轴。

axis('off')　或　axis off

axis('on')　或　axis on

例如，图 2 - 12 所示为用 axis('off')命令将图 2 - 11 所示图形去掉坐标轴后的单位圆。

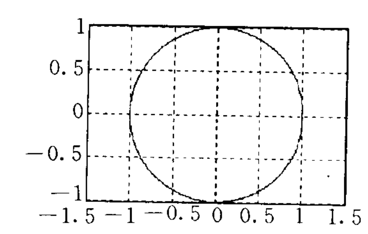


图 2 - 11　用 axis('equal')调整刻度后的单位圆图

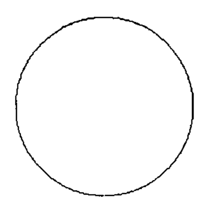


图 2 - 12　用 axis('off')去掉坐标轴后的单位圆

三、保存坐标轴的范围

· 格式：

```
variable = axis
```

· 功能：

将坐标轴的范围值[xmin xmax ymin ymax]存储到向量变量 variable 中。

· 说明：

变量 variable 保存的是一个向量值，显然这个向量值能够以 axis(variable)的形式应用于设定坐标轴的大小范围。

例如，对于由示例程序 EX201 所绘制出的图 2－9，可由下述命令来检查其坐标轴的范围。

```
》EX201
》AxisRange = axis
AxisRange =
    -1.5000   1.5000   -0.6363   0.6363
```

四、保存坐标轴的状态

· 格式：

```
[ s1 , s2 , s3 ] = axis( 'state' )
```

· 功能：

将当前所使用的坐标轴的状态存储到向量[s1 , s2 , s3]中。

· 说明：

① s1 说明是否自动设定坐标轴的范围，取值为'auto'或'manual'。

② s2 说明是否关闭坐标轴，取值为'on'或'off'。

③ s3 说明所使用的坐标轴的种类，取值为'xy'或'ij'。

例如，对于由示例程序 EX201 所绘制出的图 2－9，可由下述命令来检查其坐标轴的当前状态。

```
》EX201
》[ s1 , s2 , s3 ] = axis('state')
s1 =
manual
s2 =
on
s3 =
xy
```

2.2.2 画出或取消网格线

· 格式：

```
grid on
grid off
```

· 功能：

grid on 命令在所画出的图形中添加网格线；

grid off 命令对已有网格线的图形去掉其网格线。

例如，图 2 - 13 所示为带有网格线的图形，使用 grid off 命令可去掉其网格线，所得结果如图 2 - 14 所示。

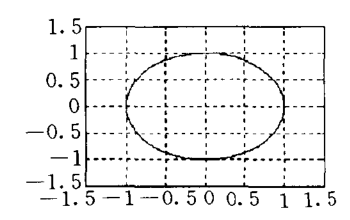


图 2 - 13　带有网格线的图形

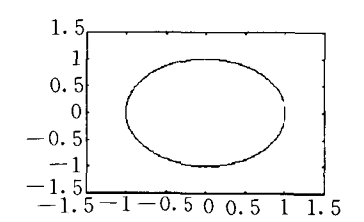


图 2 - 14　不带有网格线的图形

2.2.3　设置坐标轴名称

- 格式：

  xlabel(str)

  ylabel(str)

- 功能：

xlabel(str)命令将字符串 str 水平放置于 X 轴，以说明 X 轴数据的含义。

ylabel(str)命令将字符串 str 垂直放置于 Y 轴，以说明 Y 轴数据的含义。

2.2.4　设置图形标题

- 格式：

  title(str)

  title(variable)

- 功能：

在所画出图形的最上端，显示说明该图形标题的字符串 str，或显示某一变量 variable 的值，以该值作为所画图形的标题。

例如，下述示例程序 EX202 将绘制出一个带有坐标轴名称及图形标题的图形，如图 2 - 15 所示。

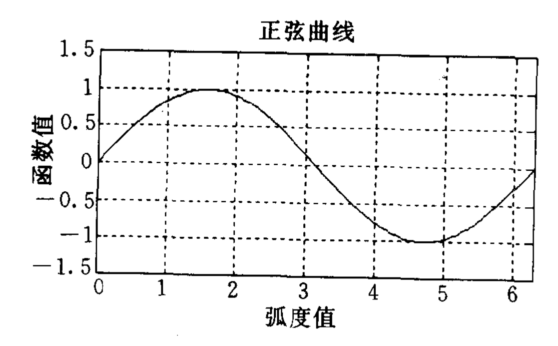


图 2 - 15　带有坐标轴名称及图形标题的正弦曲线

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX202.M
% 功能：绘制带有坐标轴名称及图形标题的正弦曲线
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = 0 : pi/100 : 2 * pi;
y = sin(x);
plot( x , y )
grid on
```

```
axis([0  2*pi  -1.5  1.5])
xlabel('弧度值')
ylabel('函数值')
title('正弦曲线')
```

2.2.5 在图形中显示文字

MATLAB 允许用户在图形的任意位置加注一串文字。加注文字时,MATLAB 提供了两种不同的确定文字位置的操作方式:用坐标值确定文字位置;用鼠标确定文字位置。

一、用坐标值确定位置显示文字

MATLAB 允许用户在图形窗口的任意位置用低级命令书写一串字符。

- 格式:

```
text( x , y , string , option )
```

- 功能:

在图形的指定坐标位置(x , y)处,写出由 string 所给出的字符串。

- 说明:

坐标(x , y)的单位是由选项参数 option 决定的。如果不给出该选项参数,则(x , y)坐标的单位与图中的单位是一致的;如果选项参数取为'sc',则(x , y)坐标表示规范化的窗口相对坐标,其变化范围为 0~1,即该窗口绘图范围的左下角坐标为(0 , 0),右上角坐标为(1 , 1)。

例如,下述示例程序 EX203 将绘制出图 2 - 16 所示图形。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX203.M
% 功能:绘制带有文字说明的正弦曲线
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = 0 : pi/100 : 2*pi;
y = sin(x);
plot( x , y )
axis([0  2*pi  -1.5  1.5])
line([0,2*pi],[0,0])
text( 0.5, 0.85 , '正弦曲线' , 'sc' )
text( 0.5*pi , 0.5 , 'positive' )
text( 1.5*pi , -0.5 , 'negative' )
```

二、用鼠标确定位置显示文字

用 text 命令可以在图形的任意位置上加注文字,但前提是必须知道其位置坐标。MATLAB 也允许用户用鼠标移动的方式在图形窗口中的某一位置放置一个字符串。

- 格式:

```
gtext( string )
```

- 功能:

利用鼠标,在图形的某一位置写出由 string 所给出的字符串。

- 说明:

在 MATLAB 的命令窗口中输入 gtext 命令后,在图中将会出现一个十字架,用鼠标拖动它到需要添加文字的地方,然后单击鼠标,就可将 gtext 命令中的字符串添加到图中。文字的

添加过程如图 2 - 17 所示。

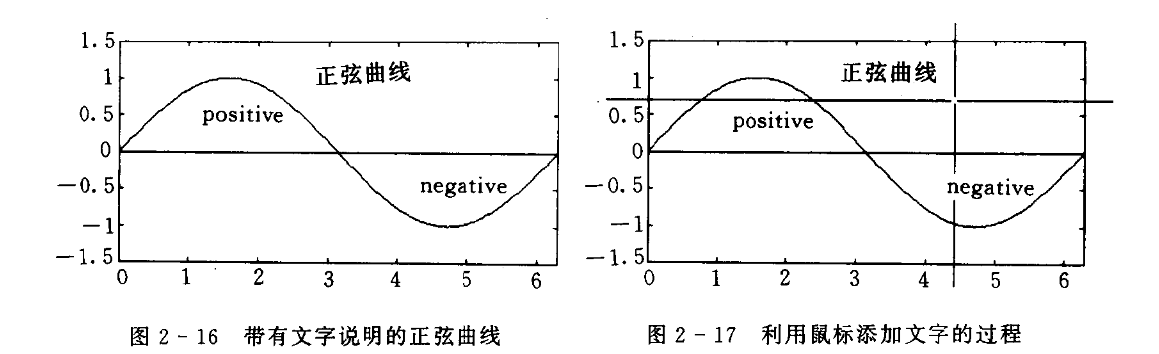


图 2 - 16　带有文字说明的正弦曲线　　　图 2 - 17　利用鼠标添加文字的过程

2.2.6　图例的标定

在 MATLAB 的图形窗口中，当在一个坐标系下画有多幅图形时，为更明确地区分各个图形，系统提供了图例的注解说明命令 legend。

- 格式：

legend(string1 , string2 , string3 , … , option)

legend(linestyle1 , string1 , linestyle2 , string2 , linestyle3 , string3 , … , option)

- 功能：

在屏幕中开启一个小视窗，然后依据绘图命令的先后次序，用对应的字符串对它们进行注解说明；或者是利用对应的字符串，对指定线型的图形进行注解说明。

- 说明：

① 参数 option 可以省略。省略时，表示将注解放置在图形视窗之内。而当 option 取值为−1 时，表示强行将注解放置在图形视窗之外。

② 注解小视窗可以用鼠标进行拖动，以将其放置到一个合适的位置。

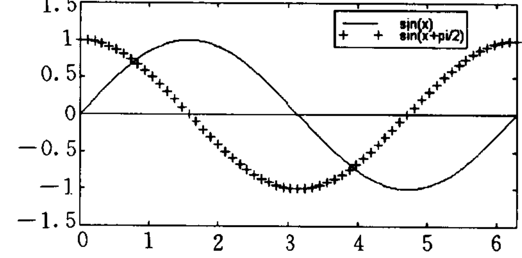


图 2 - 18　带有注解说明文字的正弦曲线

例如，下述示例程序 EX204 将绘制出图 2 - 18 所示带有注解说明的图形。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX204.M
% 功能:绘制带有注解说明文字的正弦曲线
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = 0 : pi/30 : 2 * pi;
y1 = sin(x);
y2 = sin(x+pi/2);
plot( x , y1 , x , y2 , '+' )
axis([0  2 * pi  -1.5  1.5])
line([0,2 * pi],[0,0])
```

legend('sin(x)' , 'sin(x+pi/2)')

如果将示例程序 EX204 的最后一条语句改为:

legend('sin(x)' , 'sin(x+pi/2)' , -1)

则可绘制出图 2 - 19 所示的在图形视窗之外带有注解说明的图形。

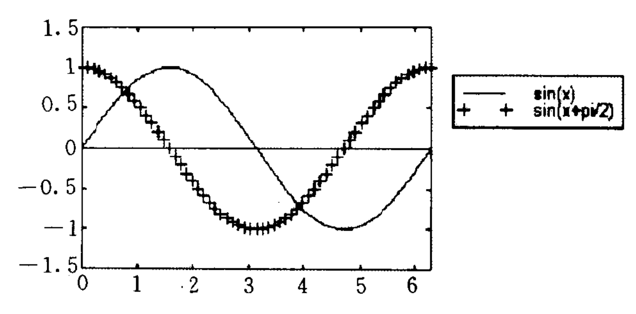


图 2 - 19 在图形视窗之外带有注解说明文字的正弦曲线

2.3 填充图形的绘制

有时,为美观起见,我们需要对 1 个封闭的图形进行填充处理。MATLAB 提供了 1 个用于图形填充的命令 fill。

· 格式:

fill (x , y , color)

· 功能:

在由数据 x , y 所构成的多边形内,填充由 color 所指定的颜色。

· 说明:

① 参数 x , y 可以是 1 个向量,也可以是 1 个矩阵。当它们是 1 个矩阵时,参数 color 就必须是 1 个列向量,其各个元素表示对应的各个 x , y 行向量所表示图形的填充颜色。

② color 参数有两种取法,一种是由'y','m','c','r','g','b','w','k'等字母来表示,其含义如表 2 - 3 所示。

表 2 - 3 颜色的修饰字母

字母	y	m	c	r	g	b	w	k
颜色	黄色	洋红色	蓝绿色	红色	绿色	蓝色	白色	黑色

另一种是由三原色向量来合成,即

color = [r g b]

其中,r,g,b 分别代表红色、绿色、蓝色所占分量的多少,它们的取值范围是 0~1。如,color = [1 0 0]代表红色,color = [0 0 1]代表蓝色,color = [1 1 1]代表白色。

例如,下述示例程序 EX205 将绘制出图 2 - 20 所示的填充图形。

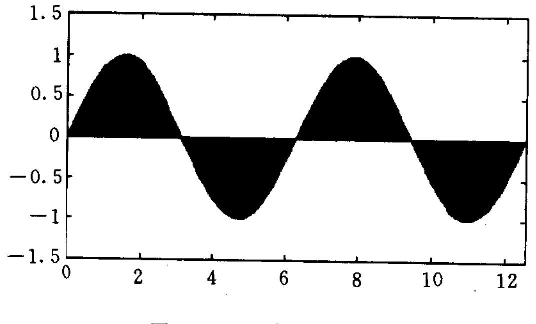


图 2 - 20 填充图形

```
% * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX205.M
% 功能:绘制填充图形
% * * * * * * * * * * * * * * * * * * * * * *
x = linspace(0,4 * pi,100);
y = sin(x);
fill( x , y , [ 0  0  0 ] )
axis( [ 0  4*pi  -1.5  1.5 ] )
```

2.4 多坐标系绘图与图形窗口的分割

在科学研究中,有时我们需要在一张图纸中绘制多幅图形,以便于观看它们之间的关系。虽然我们可以通过坐标平移的方法达到这个要求,但这种方法操作起来比较麻烦。对于这类问题,MATLAB 提供了两种解决方法:一种是使用同一坐标绘制多幅图形,即图形叠印的方法;另一种是利用图形窗口的分割实现多幅图形的绘制。

2.4.1 图形叠印

所谓图形叠印,也就是在同一坐标系中画出多幅图形。其实,前面我们介绍过的绘图命令

```
plot( x1 , y1 , option1 , x2 , y2 , option2 ,… )
```

就能在同一坐标系中画出多幅图形。

但要说明的是,plot 命令是一个一体化的绘图指令,该命令执行时将首先对当前图形窗口清屏,所以我们看到的只是最后一个 plot 命令所绘制出的图形。为了能够利用多条 plot 命令绘制出多幅图形,就必须将当前图形窗口上的图形保留住。MATLAB 提供了专门用于图形保留的命令 hold。

- 格式:

  hold on

  hold off

- 功能:

hold on　保留当前窗口中所绘制出的图形。

hold off　命令能够解除 hold on 命令。

例如,下述示例程序 EX206 中,我们使用了两条 plot 命令,它们在同一个坐标系中绘制出了两幅图形,如图 2 - 21 所示。

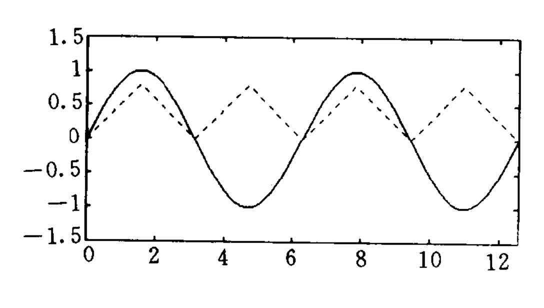


图 2 - 21　图形叠印

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX206.M
% 功能:图形叠印
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x1 = 0 : pi/200 : 4 * pi;
y1 = sin(x1);
```

```
plot( x1 , y1 )
hold on
x2 = 0 : pi/2: 4 * pi;
y2 = [ 0 , 0.8 , 0 , 0.8 , 0 , 0.8 , 0 , 0.8 , 0 ];
plot( x2 , y2 , ':' )
hold off
axis( [ 0  4 * pi  -1.5  1.5 ] )
```

2.4.2 图形窗口分割

在一个图形窗口中绘制多幅图形的另一种方法是利用图形窗口分割函数 subplot( )将当前绘图窗口分割成几个区域，然后再在各个区域中分别绘图。subplot( )函数的使用方法如下所述。

· 格式：

subplot (m , n , p)

· 功能：

将当前绘图窗口分割成 m 行、n 列，并且现在正准备在其中的第 p 个区域绘图。

· 说明：

① 各个绘图区域的编号原则是“先上后下、先左后右”，即与矩阵元素下标的排列次序一致。

② MATLAB 最多允许 9×9 的分割，并且它允许每个绘图区域都以不同的坐标系单独绘制图形。

③ 若我们在使用了 subplot 命令之后，又想使用 MATLAB 所缺省的 1 幅图 1 个图形窗口的方式，则必须执行 subplot (1, 1, 1)指令，或者执行清除窗口分割的命令 clf。

例如，下述示例程序 EX207 中，我们将图形窗口分割成 4 个区域，并分别绘制出图形，如图 2 - 22 所示。

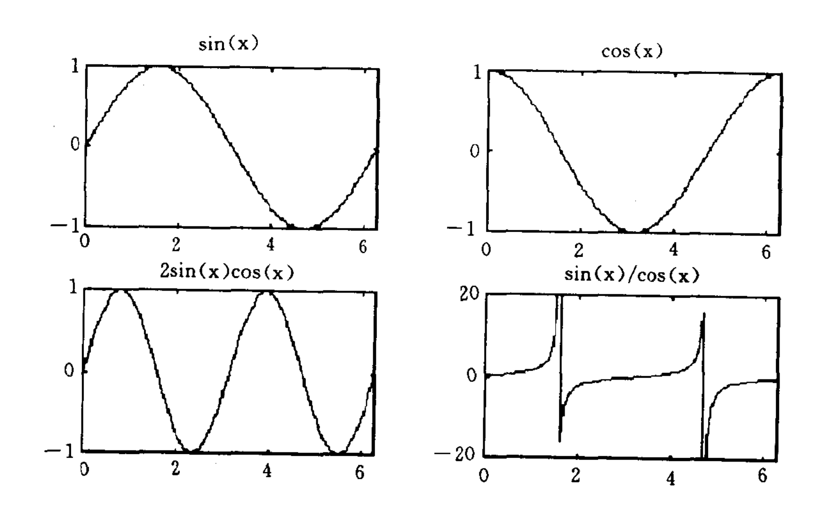


图 2 - 22 图形窗口的分割

例如，下述示例程序 EX208 可绘制出图 2 - 23 所示的半对数坐标图形，其中下边所示为用作比较的线性坐标图形。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX208.M
% 功能：半对数坐标图形与线性坐标图形的比较
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = 0 : 0.1 : 10;
y = 10. ^ x;
subplot( 2 , 1 , 1 )
semilogy( x , y )
title('Semilogarithmic Scales Graph')
grid on
subplot( 2 , 1 , 2 )
plot( x , y )
title('Linear Scales Graph')
grid on
```

2.5.2 绘制对数坐标图形

- 格式：

loglog(x , y , option)

- 功能：

本命令绘制出 X 轴和 Y 轴都为对数坐标(以 10 为底)的二维折线图形。

- 说明：

该命令的使用方法及参数含义都与 plot 命令相类似，此处不再赘述。

例如，下述示例程序 EX209 可绘制出图 2 - 24 所示的对数坐标图形，其中下边所示为用作比较半对数坐标图形。

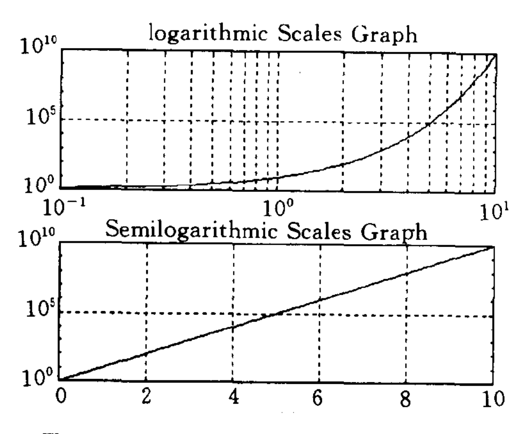


图 2 - 24 对数坐标图形与半对数坐标图形

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX209.M
% 功能：对数坐标图形与半对数坐标图形的比较
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = 0 : 0.1 : 10;
y = 10. ^ x;
subplot( 2 , 1 , 1 )
loglog( x , y )
title('logarithmic Scales Graph')
grid on
subplot( 2 , 1 , 2 )
semilogy( x , y )
```

```
title('Semilogarithmic Scales Graph')
grid on
```

2.5.3 绘制极坐标图形

• 格式：

polar (theta , radius , option)

• 功能：

绘制出一个二维折线图形，该图形的各个数据点由极坐标的形式给出。

• 说明：

① 参数 theta 表示各个数据点的角度向量。

② 参数 radius 表示各个数据点的幅值向量。

③ 参数 option 是 1 个选项参数，其含义与 plot 函数的选项参数类似。

④ 向量参数 theta 和 radius 的长度必须一致。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2－25 所示的图形。

```
》t = 0 : 0.01 : 2 * pi;
》r = 2 * sin(2 * (t-pi/8)). * cos(2 * (t-pi/8));
》polar( t , r )
```

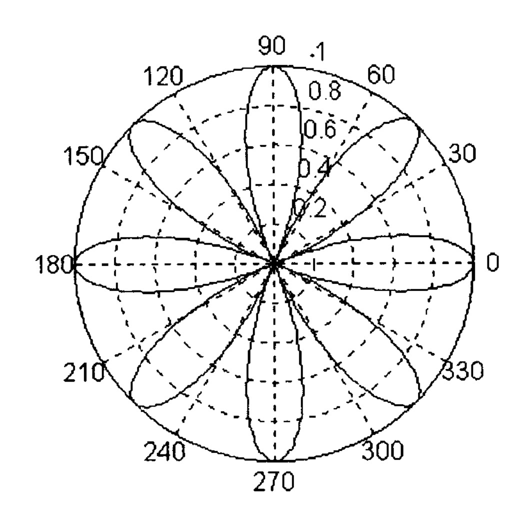


图 2－25 极坐标图形

2.6 特殊二维图形的绘制

除了折线型的图形之外，很多工程及研究领域还使用了其他一些不同类型的特殊二维图形，如统计学中的直方图、概率分布图，误差分析中的误差图等。MATLAB 也提供了绘制这些实用图形的命令。

2.6.1 圆饼图绘制命令 pie

圆饼图在统计中常用来表示各因素所占的百分比示例。在 MATLAB 中画圆饼图的命令是 pie。

· 格式：

pie (y)

pie (y, explode)

· 功能：

绘制数据 y 所定义的圆饼图，以表示各数据所占的百分比。

· 说明：

① 若 y 为向量值时，则该命令绘制出每一元素占全部向量元素总和值的百分比的圆饼图。

② 若 y 为矩阵值时，则该命令绘制出每一元素占全部矩阵元素总和值的百分比的圆饼图。

③ 参数 explode 说明是否将某一数据对应的扇形图形从整个圆饼图中分离出来，它的维数与 y 相同，当它的某个元素非零时，即表示将对应的扇形图形从整个圆饼图中分离出来。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2 - 26 所示的圆饼图。

```
》y = [ 15 , 35 , 10 , 20 , 20 ];
》subplot(1,2,1)
》pie(y)
》subplot(1,2,2)
》pie( y ,[1,0,0,1,0] )
```

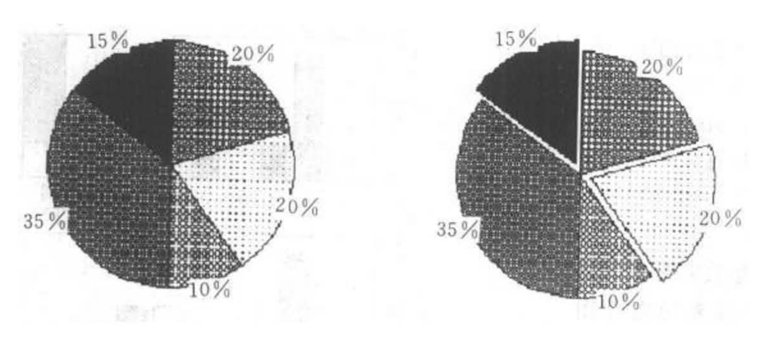


图 2 - 26 圆饼图示例

2.6.2 直方图绘制命令 bar

直方图可用来表示一些数据的对比情况。MATLAB 提供了两类画直方图的命令，一类是用于画垂直放置的直方图，另一类是用于画水平放置的直方图。

一、画垂直放置的直方图

· 格式：

bar(y , option)

```
bar( x , y , option )
bar( y , 'stack' )
bar( y , 'group' )
```

• 功能：

① 命令格式 bar(y , option)以 x=1,2,3,…为各个数据点的 x 坐标，以 **y** 向量的各个对应元素为 y 坐标，画出一个垂直放置的二维直方图。

② 命令格式 bar(x , y , option)以向量 **x** 的各个对应元素为 x 坐标，以 **y** 向量的各个对应元素为 y 坐标，画出一个垂直放置的二维直方图。

③ 命令格式 bar(y , 'stack')以 x=1,2,3,…为各个数据点的 x 坐标，以矩阵 **y** 的各个列向量的累加值为 y 坐标，画出 1 个垂直放置的、累加式的二维直方图。

④ 命令格式 bar(y , 'group')以 x=1,2,3,…为各个数据点的 x 坐标，以矩阵 **y** 的各个列向量的累加值为 y 坐标，画出 1 个垂直放置的、分组式的二维直方图。

• 说明：

① 如果 x,y 同为维数相同的矩阵，则该命令将以 **x**,**y** 的每一个行向量为数据，画出一组直方图。

② 字符串参数 option 表示绘图时的线型、颜色，其取值见表 2-1。

例如，下述示例程序 EX210 可绘制出图 2-27 所示的几个不同类型的直方图。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX210.M
% 功能:直方图绘制示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
y1 = [ 15 , 35 , 10 , 20 , 20 ];
y2 = [ 15 , 35 , 10 ; 20 , 20 , 15 ; 10 , 15 , 30 ];
subplot( 2 , 2 , 1 )
bar( y1 )
title('单个直方图')
subplot( 2 , 2 , 2 )
bar( y2 , 'stack' )
title('累加直方图')
subplot( 2 , 1 , 2 )
bar( y2 )
title('分组直方图')
```

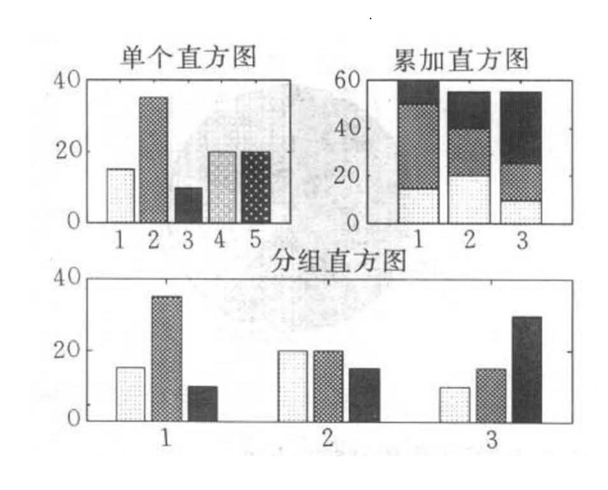


图 2-27　直方图绘制示例

二、画水平放置的直方图

• 格式：

```
barh( y , option )
barh( x , y , option )
barh( y , 'stack' )
```

• 功能：

这 3 个命令的功能及使用方法与前述 3 个 bar 命令的功能及使用方法相同，它们的区别在于所画出的直方图是水平放置的，而不是垂直放置的。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2-28 所示的水平放置的直

方图。

```
》y = [ 15 , 35 , 10 , 20 , 20 ];
》barh(y)
```

三、直方图数据的记忆

- 格式:

  [xdata , ydata] = bar(x , y)

- 功能:

由 x ，y 所定义的直方图产生一对向量值 xdata,ydata(并不画出直方图),该对向量值可利用 plot(xdata , y)命令再次画出直方图。

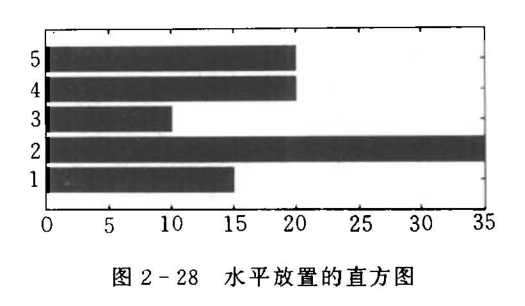


图 2-28 水平放置的直方图

2.6.3 区域图绘制命令 area

- 格式:

  area(y)

  area(x , y)

- 功能:

① 命令格式 area(y)以 x=1,2,3,…为各个数据点的 x 坐标,以 y 向量的各个对应元素为 y 坐标,画出 1 个二维折线图,并填充该折线与 X 轴之间的区域。

② 命令格式 area(x , y)以向量 x 的各个对应元素为 x 坐标,以 y 向量的各个对应元素为 y 坐标,画出 1 个二维折线图,并填充该折线与 X 轴之间的区域。

例如,在 MATLAB 命令窗口中键入下述命令,可画出图 2-29 所示的区域图。

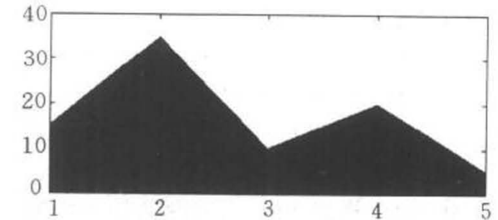


图 2-29 区域图示例

```
》y = [ 15 , 35 , 10 , 20 , 5 ];
》area(y)
```

2.6.4 阶梯图绘制命令 stairs

阶梯图可用来表示系统中的采样数据。

一、阶梯图的绘制

- 格式:

  stairs(y)

  stairs(x , y)

- 功能:

① 命令格式 stairs(y)以 x=1,2,3,…为各个数据点的 x 坐标,以 y 向量的各个对应元素为 y 坐标,画出 1 个阶梯状的图形。

② 命令格式 stairs(x , y)以向量 *x* 的各个对应元素为 x 坐标，以 *y* 向量的各个对应元素为 y 坐标，画出 1 个阶梯状的图形。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2 - 30 所示的阶梯图。

```
》x = 0 : pi/20 : 4 * pi;
》y = abs(sin(x));
》stairs(x,y)
》axis( [ 0   4 * pi   -0.5   1.5 ] )
```

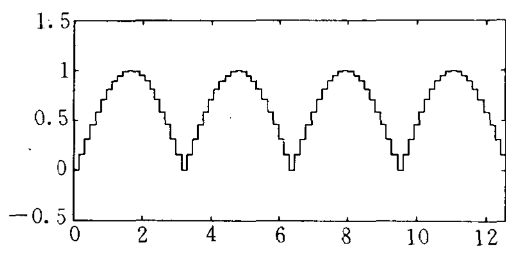


图 2 - 30 阶梯图示例

二、阶梯图数据的记忆

- 格式：

  [xdata , ydata] = stairs(x , y)

- 功能：

由 x , y 所定义的阶梯图产生一对向量值 xdata 及 ydata(并不画出这个阶梯图)，该对向量值数据可利用 plot(xdata , ydata)命令再次画出阶梯图。

2.6.5 概率分布图绘制命令 hist

概率分布图是研究随机系统时经常要用到的一种描述图形，它有很广泛的用途。

一、概率分布图的绘制

- 格式：

  hist(y)

  hist(y , n)

  hist(y , x)

- 功能：

① 命令格式 hist(y)将向量 *y* 的最大值与最小值的差平均分成 10 等分，然后绘制出其分布图。

② 命令格式 hist(y , n) 将向量 *y* 的最大值与最小值的差平均分成 n 等分，然后绘制出其分布图。

③ 命令格式 hist(y , x)以向量 *x* 的各个元素值为统计范围，绘制 *y* 的分布图。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2 - 31 所示的概率分布图。

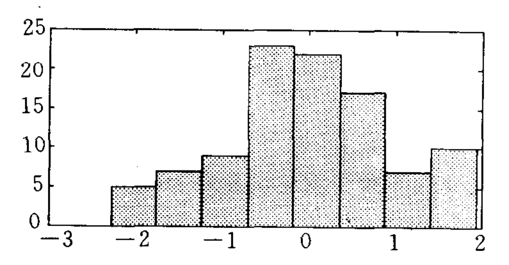


图 2 - 31 概率分布图示例

```
》randn( 'seed' , sum(clock) );
》y = randn( 1 , 100 );
》hist( y , 8 )
```

其中的第 1 条 randn 命令中的'seed'项，说明了如何产生随机数的起始种子数。本例中，我们取时钟向量 clock[yy   mm   dd   hh   mm   ss]的和为起始种子数。

二、概率分布图数据的记忆

· 格式：

[xdata , ydata] = hist(x , y)

· 功能：

由 x ，y 所定义的概率分布图产生一对向量值 xdata，ydata(并不画出这个概率分布图)，该对向量值可利用 bar(xdata , ydata)命令画出直方图形式的概率分布图。

2.6.6 条形误差图绘制命令 errorbar

条形误差图可用来表示各种数据相对于理想数据的偏差情况。

· 格式：

errorbar(x , y , L , U , option)

errorbar(x , y , e)

· 功能：

① 命令 errorbar(x , y , L , U , option)以 x 数据为 X 轴，以 y 数据为 Y 轴，先绘制出一条曲线图，然后在对应的各个(x,y)数据点处画一垂直的线段，其向下的长度由向量 ***L*** 的对应元素来定义，向上的长度由向量 ***U*** 的对应元素来定义。

② 命令 errorbar(x , y , e)以 x 数据为 X 轴，以 y 数据为 Y 轴，先绘制出一条曲线图，然后在对应的各个(x,y)数据点处画一垂直的线段，其向上和向下的长度是相等的，长度均为向量 *e* 的对应元素所定义。

· 说明：

① 参数 x , y , L , U , e 皆应是长度相同的向量或维数相同的矩阵。

② 字符串参数 option 是 1 个可选参数，它表示绘制曲线图时的线型、颜色，其取值见表 2 - 1。

例如，下述示例程序 EX211 可绘制出图 2 - 32 所示的条形误差图。

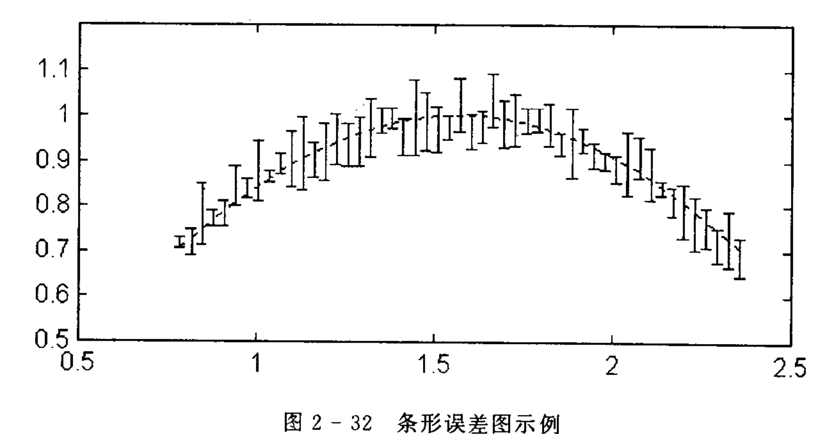


图 2 - 32 条形误差图示例

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX211.M
% 功能：条形误差图绘制示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
x = 0.25 * pi : pi/100 : 0.75 * pi;
y = sin(x);
rand('seed' , sum(clock));
Lerror = rand(1,length(x))/10;
Uerror = rand(1,length(x))/10;
errorbar( x , y , Lerror , Uerror,':')
axis( [ 0.5  2.5  0.5  1.2 ] )
```

2.6.7 离散序列图绘制命令 stem

在现代科学研究中，我们更多地要处理一些离散量。离散序列图就可以表示离散量的变化情况。

- 格式：

```
stem( y )
stem( x , y , option )
stem( x , y , 'filled' )
```

- 功能：

① 命令 stem(y)以 x=1,2,3,…为各个数据点的 x 坐标，以 ***y*** 向量的各个对应元素为 y 坐标，在(x , y)坐标点画一个空心的小圆圈，并连接一条线段到 X 坐标轴。

② stem(x , y , option)以向量 ***x*** 的各个元素为 x 坐标，以 ***y*** 向量的各个对应元素为 y 坐标，在(x , y)坐标点画一个空心的小圆圈，并连接一条线段到 X 坐标轴。

③ stem(x , y , 'filled')以向量 ***x*** 的各个元素为 x 坐标，以 ***y*** 向量的各个对应元素为 y 坐标，在(x , y)坐标点画一个实心的小圆圈，并连接一条线段到 X 坐标轴。

- 说明：

① 字符串参数 option 是 1 个可选参数，它表示绘制图形时的线型、颜色，其取值见表 2 - 1。

② 小圆圈的填充颜色也是由参数 option 来说明的。

例如，下述示例程序 EX212 可绘制出图 2 - 33 所示的离散序列图。

```
% * * * * * * * * * * * * * * *
% 程序:EX212.M
% 功能:离散序列图绘制示例
% * * * * * * * * * * * * * * *
x = 0 : pi/15 : 2 * pi;
y = sin(x);
stem( x , y )
hold on
x = 2 * pi : pi/15 : 4 * pi;
y = sin(x);
stem( x , y , 'filled' )
hold off
```

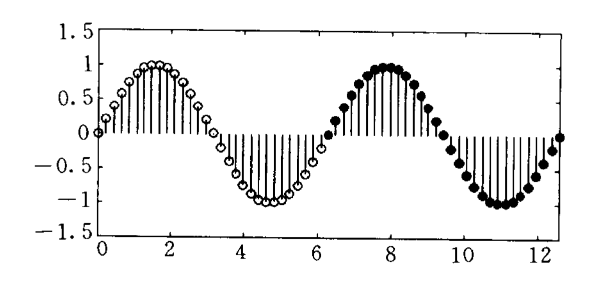


图 2 - 33 离散序列图示例

```
axis( [ 0   4 * pi   -1.5   1.5 ])
```

2.6.8 极坐标系下的概率分布图绘制命令 rose

一、极坐标系下概率分布图的绘制

- 格式：

```
rose( theta )
rose( theta , n )
rose( theta , x )
```

- 功能：

① 命令格式 rose(theta)将向量 theta 的最大值与最小值的差平均分成 20 等分，然后绘制出其分布图。

② 命令格式 rose(theta , n) 将向量 theta 的最大值与最小值的差平均分成 n 等分，然后绘制出其分布图。

③ 命令格式 rose(theta , x)以向量 *x* 的各个元素值为统计范围，绘制 theta 的分布图。

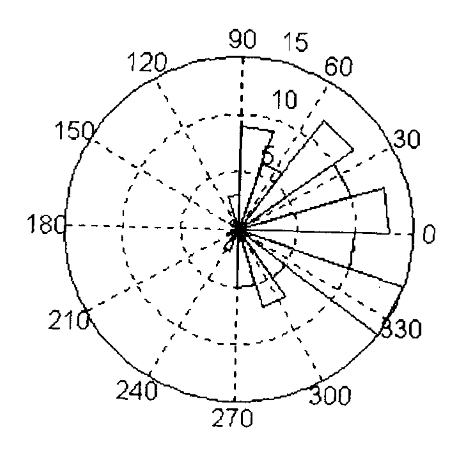


图 2-34 极坐标系下的概率分布图示例

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2-34 所示的极坐标系下的概率分布图。

```
» randn( 'seed' , sum(clock) );
» alpha = randn( 1 , 100 );
» rose( alpha )
```

二、极坐标系下概率分布图数据的记忆

- 格式：

```
[ thetadata , xdata ] = rose( theta , x )
```

- 功能：

由 theta , x 所定义的极坐标系下的概率分布图产生一组向量值 thetadata，xdata(并不画出这个概率分布图)，该组向量值可利用 polar (thetadata , xdata)命令画出其相应的概率分布图。

2.6.9 复数向量绘制命令 compass

工程计算中常用到复数向量，MATLAB 提供了绘制复数向量的命令。

- 格式：

```
compass( z , option )
compass( x , y , option )
```

- 功能：

① 命令 compass(z , option)以复坐标系的原点为起点，绘制出带有箭头的一组复数向量 *z*，既表示向量 *z* 中各个复数的大小，又表示其角度。

② 命令 compass(x , y , option)以复坐标系的原点为起点，绘制出带有箭头的一组复数

向量，其中向量 x 表示复数的实部，向量 y 表示复数的虚部。该命令也就是相当与执行命令 compass(x+j * y , option)。

• 说明：

字符串参数 option 是一个可选参数，它表示绘制复数图形时的线型、颜色，其具体取值见表 2 - 1。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2 - 35 所示的一组复数向量。

```
》z = [ 1+2j  2+2j  1-3j  3j  -3 ]
》compass(z)
```

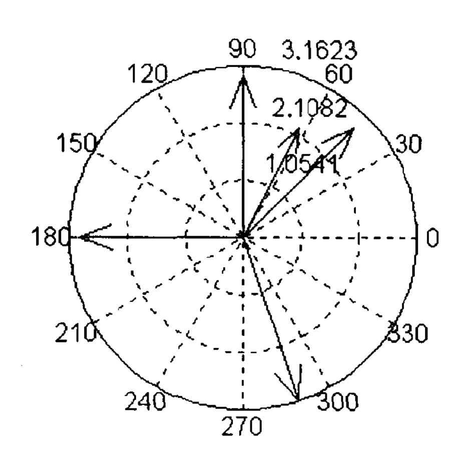


图 2 - 35 复数向量图示例

2.6.10 复数向量绘制命令 feather

• 格式：

feather(z , option)

feather(x , y , option)

• 功能：

绘制一组复数向量。

• 说明：

① 命令 feather 与命令 compass 的功能及使用方法相类似，两者的区别在于它们所绘制出的各个复数的起点不同。命令 compass 绘制的各个复数的起点都在坐标系的原点，而命令 feather 绘制的各个复数的起点却是在水平坐标轴上等间隔分布的。

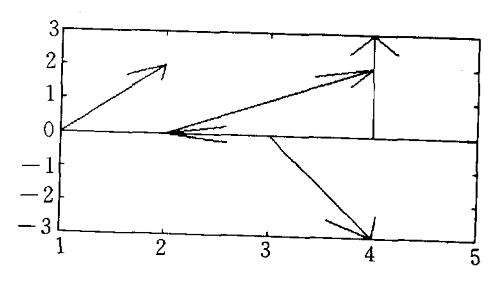


图 2 - 36 复数向量图示例

② 命令 feather 的各个参数与命令 compass 的各个参数的含义相同。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 2 - 36 所示的一组复数向量。

```
》z = [ 1+2j  2+2j  1-3j  3j  -3 ]
》feather(z)
```

2.7 手工绘图方法

手工绘图的关键是如何选取绘图时的一些关键数据点，如何将这些数据点的坐标值读入变量，然后加以利用。MATLAB 提供了用鼠标选取数据点的命令 ginput，当在图形窗口中的某一位置按下某一个鼠标键(或键盘上除回车键之外的任何键)时，ginput 将返回该位置的坐标值。

· 格式：

[x , y] = ginput

[x , y] = ginput(n)

[x , y , button] = ginput(n)

· 功能：

① 利用命令[x , y] = ginput，当在图形窗口中按下某一个鼠标键或某一键盘键时，读取此时鼠标所在位置的一系列坐标值，并将这些坐标值存储到向量 x 和 y 中，直到按回车键后才终止该读数过程。

② 利用命令[x , y] = ginput(n)，当在图形窗口中按下某一个鼠标键或某一键盘键时，读取此时鼠标所在位置的一系列坐标值，并将这些坐标值存储到向量 x 和 y 中，总共读取 n 个数据点。

③ 命令[x , y , button] = ginput(n)也可利用鼠标从图形窗口中读取 n 个数据点，并将这些数据点的坐标值存储到向量 x 和 y 中，同时还将读数过程中鼠标的按键情况或键盘的按键情况记录到向量变量 button 中。

· 说明：

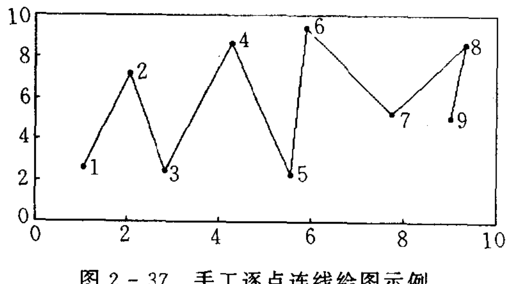
① 在读取第 i 个数据点时，若按的是鼠标左键，则 button(i)=1；若按的是鼠标中键，则 button(i)=2；若按的是鼠标右键，则 button(i)=3；若按的是键盘键，则 button(i)存储相应键的 ASCII 码。

② 该命令仅仅只是读取了一些数据点，并没有绘制图形。在读取了一些数据点之后，我们就可以利用某种方法或按某种绘图要求把这些数据点连成一起，从而达到手工绘图的效果。

例如，下述示例程序 EX213 可依据用户用鼠标选取的一些数据点来绘制折线图(如图 2-37 所示)。绘图过程中，用鼠标左键选取数据点，当按鼠标右键后，就停止选点，并用逐点连线的方法绘制图形。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX213.M
% 功能：手工逐点连线绘图示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
clf
axis( [ 0  10  0  10 ] )
hold on
```

```
x = [ ];
y = [ ];
n = 0;
while (1)
    [ xtemp , ytemp , button ] = ginput(1);
    plot( xtemp, ytemp, '.' )
    x = [ x , xtemp ];
    y = [ y , ytemp ];
    n = n + 1;
    text( xtemp+0.1 , ytemp , int2str(n) );
    if ( button == 3 )
        break
    end
end
line( x , y )
hold off
```



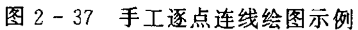


图 2－37　手工逐点连线绘图示例

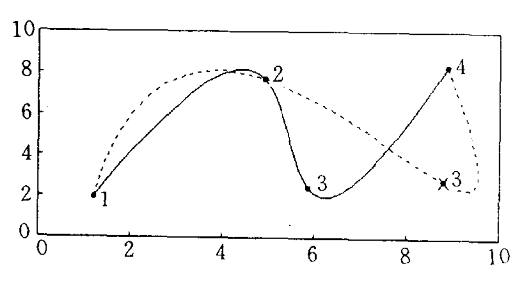


图 2－38　样条曲线的绘制及其数据点的更改

又例如，下述示例程序 EX214 可绘制出图 2－38 所示的样条曲线图。在绘制过程中，我们先依据由鼠标位置所读出的数据点绘制出一根样条曲线（如图中虚线所示），然后更改其中的某一个选定的数据点（用鼠标确定更改哪个数据点，并确定新的数据点的位置），最后再依据新的数据点绘制出一根新的样条曲线（如图中实线所示）。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX214.M
% 功能:样条曲线的绘制及其数据点的更改
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
clf
axis( [ 0   10   0   10 ] )
hold on
% read data from screen
x = [ ];
y = [ ];
n = 0;
while ( 1 )
    [ xtemp , ytemp , button ] = ginput(1);
```

```
    plot( xtemp, ytemp, '.' )
    x = [ x , xtemp ];
    y = [ y , ytemp ];
    n = n + 1;
    text( xtemp+0.1 , ytemp , int2str(n) );
    if ( button == 3 )
        break
    end
end
% draw a spline
t = 1:n;
tt = 1:0.1:n;
xx = spline( t , x , tt );
yy = spline( t , y , tt );
plot( xx , yy , 'b:' )
% judge a point
[ xtemp , ytemp , button ] = ginput(1);
for i = 1:n
    if ( (abs(x(i)-xtemp)<0.1) & (abs(y(i)-ytemp)<0.1) )
        k = i;
        line([x(k)-0.1,x(k)+0.1],[y(k)-0.3,y(k)+0.3])
        line([x(k)+0.1,x(k)-0.1],[y(k)-0.3,y(k)+0.3])
        break
    end
end
[ xtemp , ytemp , button ] = ginput(1);
plot( xtemp, ytemp, 'r.' )
x(k) = xtemp;
y(k) = ytemp;
text( xtemp+0.1 , ytemp , int2str(k) );
% draw a new spline
xx = spline( t , x , tt );
yy = spline( t , y , tt );
plot( xx , yy , 'r-' )
hold off
```

第3章　三维图形绘制

我们生活在三维空间中，现实中所遇到的一些问题，特别是科学计算及工程应用中的一些问题，往往都可以抽象为三维空间中的问题。前一章所介绍的二维图形，不便于反映三维空间的实际情况，所以在实际工作中有时需要绘制出三维图形，而且三维图形看起来更直观，也更美观。

本章介绍 MATLAB 提供的一些三维绘图命令及其使用方法，具体包括以下内容：

- 三维折线及曲线的绘制方法
- 三维网格曲面的绘制方法
- 三维阴影曲面的绘制方法
- 三维图形的显示控制
- 特殊三维图形的绘制方法

3.1　三维折线及曲线的绘制

3.1.1　三维折线及曲线的基本绘图命令

绘制二维折线或曲线时，我们可以使用 plot 命令。与这条命令类似，MATLAB 也提供了 1 个绘制三维折线或曲线的基本命令 plot3。

· 格式：

```
plot3( x1 , y1 , z1 , option1 , x2 , y2 , z2 , option2 , … )
```

· 功能：

以 x1 , y1 , z1 所给出的数据分别为 x , y , z 坐标值，option1 为选项参数，以逐点连折线的方式绘制 1 个三维折线图形；同时，以 x2 , y2 , z2 所给出的数据分别为 x , y , z 坐标值，option2 为选项参数，以逐点连折线的方式绘制另 1 个三维折线图形……

· 说明：

① plot3 命令的功能及使用方法与 plot 命令的功能及使用方法相类似，它们的区别在于前者绘制出的是三维图形。

② plot3 命令的参数含义与 plot 命令的参数含义相类似，它们的区别在于前者多了 1 个 Z 方向上的参数。同样，各个参数的取值情况及其操作效果也与 plot 命令相同。上面给出的 plot3 命令格式是一种完整的格式，在实际操作中，根据各个数据的取值情况，均可以有下述一些较简单的书写格式：

plot3(x , y , z)

plot3(x , y , z , option)

③ 选项参数 option 指明了所绘图形中线条的线型、颜色以及各个数据点的表示记号。其具体取值见表 2-1。

④ plot3 命令使用的是以逐点连线的方法来绘制三维折线的，当各个数据点的间距较小时，我们也可利用它来绘制三维曲线。

例如，在 MATLAB 命令窗口中键入下述命令，可画出图 3-1 所示的三维螺旋线。

```
» t = 0 : pi/50 : 8 * pi;
» x = sin(t);
» y = cos(t);
» z = t;
» plot3( x , y , z )
```

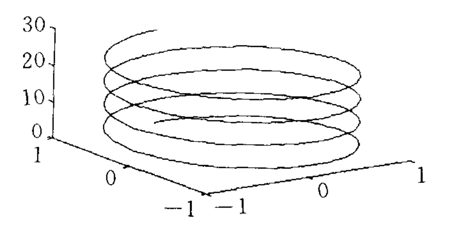


图 3-1 三维螺旋线

又例如，在 MATLAB 命令窗口中键入下述命令，可画出图 3-2 所示的三维线条图。

```
» [ X , Y ] = meshgrid( [ -3 : 0.2 : 3 ] );
» Z = X. * exp(-X. ^ 2-Y. ^ 2);
» plot3( X , Y , Z , 'b' )
```

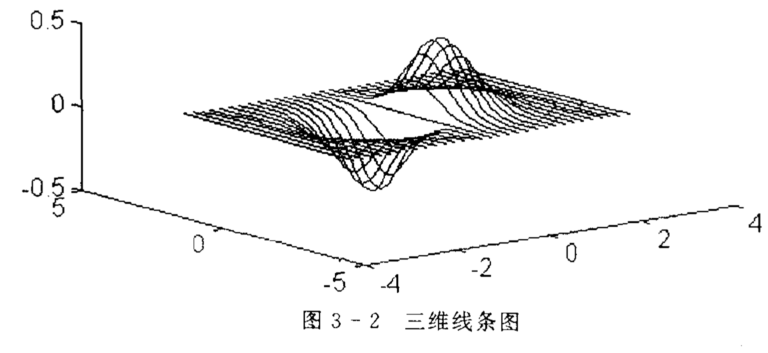


图 3-2 三维线条图

3.1.2 三维图形的坐标标记及图形标题

与二维图形的坐标标记命令类似，MATLAB 也提供了下述 3 条用于三维图形坐标标记的命令，并提供了用于图形标题说明的语句。

- 格式：

  xlabel(str)

  ylabel(str)

  zlabel(str)

  title(str)

- 功能：

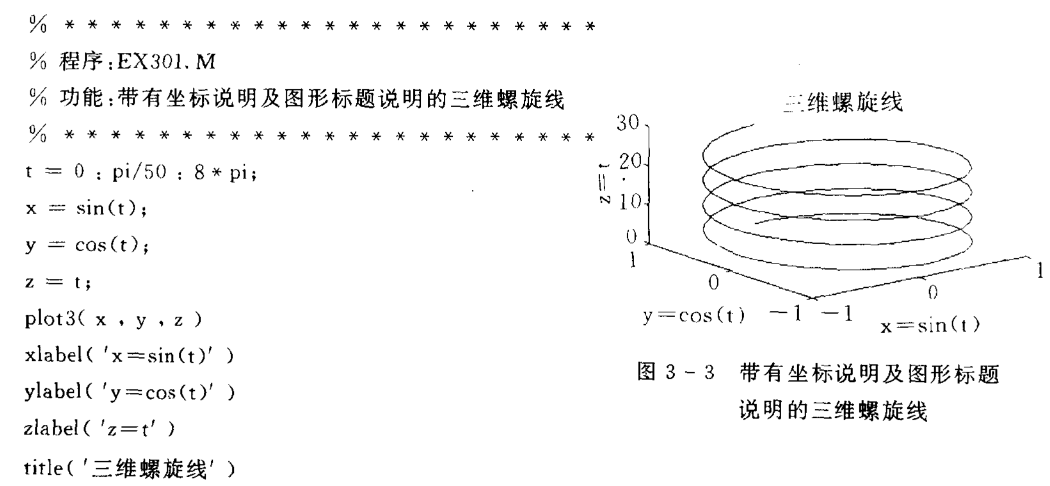
xlabel(str)命令将字符串 str 水平放置于 X 轴，以说明 X 轴数据的含义。

ylabel(str)命令将字符串 str 水平放置于 Y 轴，以说明 Y 轴数据的含义。

zlabel(str)命令将字符串 str 自下而上垂直放置于 Z 轴，以说明 Z 轴数据的含义。

title(str)命令将字符串 str 水平放置于图形的顶部，以说明该图形的标题。

例如，下述示例程序 EX301 可绘制出图 3 - 3 所示的带有坐标说明及图形标题说明的三维螺旋线。



```
% * * * * * * * * * * * * * * * * * * * * * * * *
% 程序：EX301.M
% 功能：带有坐标说明及图形标题说明的三维螺旋线
% * * * * * * * * * * * * * * * * * * * * * * * *
t = 0 : pi/50 : 8 * pi;
x = sin(t);
y = cos(t);
z = t;
plot3( x , y , z )
xlabel( 'x=sin(t)' )
ylabel( 'y=cos(t)' )
zlabel( 'z=t' )
title( '三维螺旋线' )
```


图 3 - 3 带有坐标说明及图形标题说明的三维螺旋线

3.2 三维网格曲面的绘制

三维网格曲面是由一些四边形相互连接在一起所构成的一种曲面，这些四边形的 4 条边所围区域的颜色与图形窗口的背景色相同，并且无色调的变化，呈现的是一种线架图的形式。

绘制这种网格曲面图时，我们需要知道各个四边形的顶点的(x , y , z)3 个坐标值，然后再使用 MATLAB 所提供的网格曲面绘图命令 mesh，meshc 或 meshz 来绘制不同形式的网格曲面。

3.2.1 栅格数据点的产生

前面我们讲过，在绘制网格曲面之前，必须先知道各个四边形顶点的三维坐标值。绘制曲面的一般情况是，我们先知道四边形各个顶点的二维坐标(x , y)，然后再利用某个函数公式计算出四边形各个顶点的 z 坐标。这里所使用的(x , y)二维坐标值是一种栅格形的数据点，它可由 MATLAB 所提供的 meshgrid 命令产生。

• 格式：

```
[ X , Y ] = meshgrid( x , y )
```

• 功能：

由 **x** 向量和 **y** 向量值通过复制的方法产生绘制三维图形时所需的栅格数据 **X** 矩阵和 **Y** 矩阵。

• 说明：

① 向量 **x** 和 **y** 分别代表三维图形在 X 轴、Y 轴方向的取值数据点。

② x 和 y 分别是 1 个向量，而 X 和 Y 分别代表 1 个矩阵。该命令产生栅格数据的方法如图 3 - 4 所示。即将向量 **x** 作为矩阵 **X** 的 1 个行向量，并将向量 **x** 复制 length(y)次，以构成栅

格形的数据点 *X* 矩阵。同样，将向量 *y* 作为矩阵 *Y* 的一个列向量，并将向量 *y* 复制 length(x) 次，以构成栅格形的数据点 *Y* 矩阵，如图 3－4 所示。

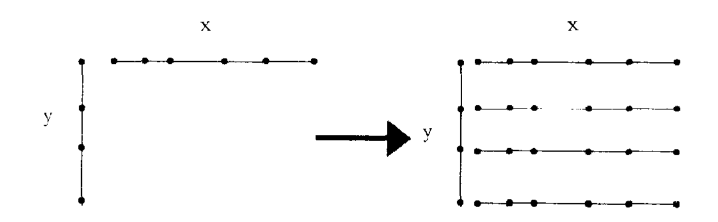


图 3－4　栅格数据点的产生方法

例如：

```
》x = [ 1   2   3   4   5   6   7   8 ];
》y = [ 3   5   7 ];
》[ X , Y ] = meshgrid( x , y )
X =
     1   2   3   4   5   6   7   8
     1   2   3   4   5   6   7   8
     1   2   3   4   5   6   7   8
Y =
     3   3   3   3   3   3   3   3
     5   5   5   5   5   5   5   5
     7   7   7   7   7   7   7   7
```

(3) 在 meshgrid 命令的参数中，若 *x* 向量和 *y* 向量的取值相同，则可省去 1 个参数，此时的命令格式为：

[X , Y] = meshgrid(x)

它相当于执行下述命令：

[X , Y] = meshgrid(x , x)

3.2.2　网格曲面的绘制命令 mesh

- 格式：

mesh(X , Y , Z , C)
mesh(X , Y , Z)
mesh(x , y , Z , C)
mesh(x , y , Z)
mesh(Z , C)
mesh(Z)

- 功能：

绘制 1 个三维的网格曲面图。

· 说明：

上述 6 种命令格式都可绘制出三维网格曲面图，但各个格式的命令参数含义有些区别，现说明如下：

① 在命令格式 mesh(X , Y , Z , C)和 mesh(X , Y , Z)中，参数 X , Y , Z 都为矩阵值，并且 ***X*** 矩阵的每一个行向量都是相同的，***Y*** 矩阵的每一个列向量也都是相同的。参数 C 表示网格曲面的颜色分布情况，若省略该参数则表示网格曲面的颜色分布与 Z 方向上的高度值成正比。

② 在命令格式 mesh(x , y , Z , C)和 mesh(x , y , Z)中，参数 x 和 y 为长度分别是 n 和 m 的向量值，而参数 Z 是维数为 m×n 的矩阵。其实，这种格式的命令相当于执行了下面两条命令：

```
[ X , Y ] = meshgrid( x , y )
mesh( X , Y , Z , C )
```

③ 在命令格式 mesh(Z , C)和 mesh(Z)中，若参数 Z 是维数为 m×n 的矩阵，则绘图时的栅格数据点的取法是：x = 1:n;y = 1:m。其实，这种格式的命令相当于执行了下面 5 条命令：

```
[ m , n ] = size(Z);
x = 1:n;
y = 1:m;
[ X , Y ] = meshgrid( x , y );
mesh( X , Y , Z , C )
```

例如，下述示例程序 EX302 可绘制出图 3 - 5 所示的网格曲面图，这个网格曲面图是下述函数的图形

$$f(x,y)=\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$$

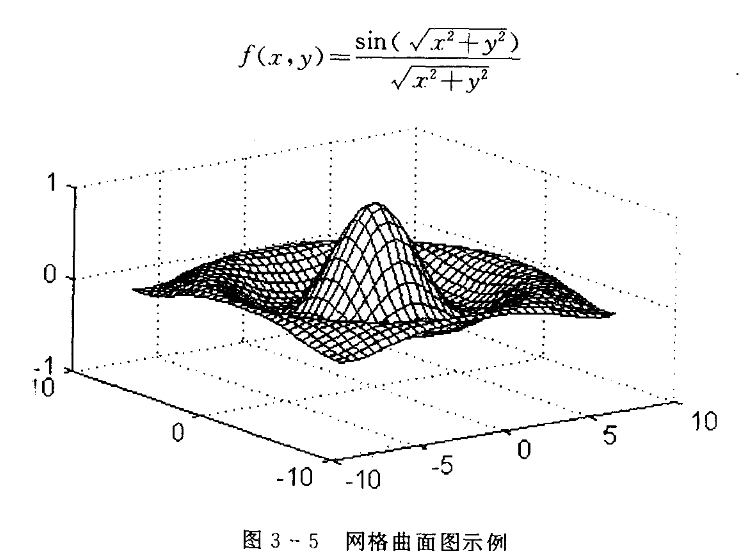


图 3 - 5　网格曲面图示例

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX302.M
% 功能:网格曲面图的绘制
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X. ^ 2 + Y. ^ 2 ) + eps;
Z = sin(R). /R;
mesh( X , Y , Z )
grid on
axis([ -10  10  -10  10  -1  1 ])
```

3.2.3 带有等高线的网格曲面绘制命令 meshc

• 格式:

```
meshc( X , Y , Z , C )
meshc( X , Y , Z )
meshc( x , y , Z , C )
meshc( x , y , Z )
meshc( Z , C )
meshc( Z )
```

• 功能:

绘制在 XY 平面上带有等高线的三维网格曲面。

• 说明:

① 这 6 个 meshc 命令与相应的 6 个 mesh 命令的使用方法及参数含义相同。

② meshc 命令与 mesh 命令的区别是前者除了绘制出三维网格曲面外,在 XY 坐标平面上还绘制有曲面在 Z 轴方向上的等高线。

例如,下述示例程序 EX303 可绘制出图 3-6 所示的函数 $f(x,y)=\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$ 的网格曲面,这个网格曲面的下面还带有函数的等高线。

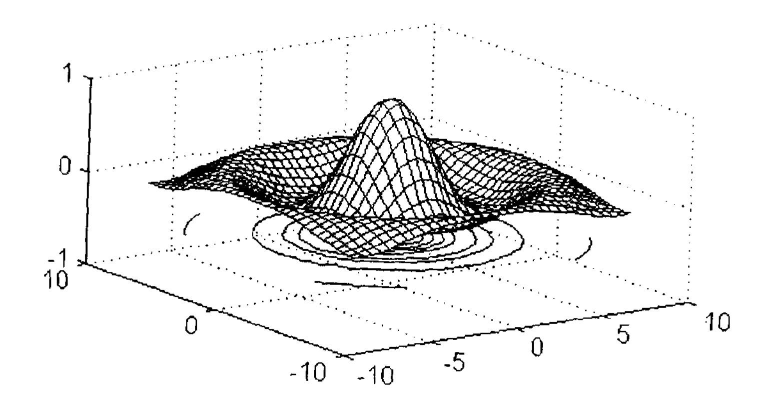


图 3-6 带有函数等高线的网格曲面图示例

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX303.M
% 功能:带有函数等高线的网格曲面图的绘制
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X.^2 + Y.^2 ) + eps;
Z = sin(R)./R;
meshc( X , Y , Z )
grid on
axis([ -10  10  -10  10  -1  1 ])
```

3.2.4 带有底坐的网格曲面绘制命令 meshz

· 格式:

```
meshz( X , Y , Z , C )
meshz( X , Y , Z )
meshz( x , y , Z , C )
meshz( x , y , Z )
meshz( Z , C )
meshz( Z )
```

· 功能:

绘制 1 个带有底座的三维网格曲面。

· 说明:

① 这 6 个 meshz 命令与相应的 6 个 mesh 命令的使用方法及参数含义相同。

② meshz 命令与 mesh 命令的区别是前者除了绘制出三维网格曲面外,另外还绘制有曲面的底座。

例如,下述示例程序 EX304 可绘制出图 3-7 所示的函数 $f(x,y)=\dfrac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$的网格曲面,这个网格曲面的下面还带有函数的底座。

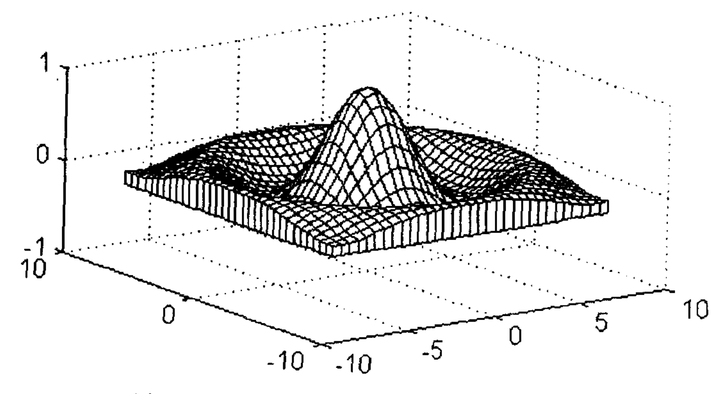


图 3-7 带有函数底座的网格曲面图示例

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX304.M
% 功能:带有函数底座的网格曲面图的绘制
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X.^2 + Y.^2 ) + eps;
Z = sin(R)./R;
meshz( X , Y , Z )
grid on
axis([ -10  10  -10  10 , -1  1 ])
```

3.2.5 隐藏线的显示与关闭

显示或不显示网格曲面的隐藏线将对图形的显示效果有一定的影响。MATLAB 提供了相关的控制命令 hidden。

- 格式:

  hidden on

  hidden off

- 功能:

hidden on　命令去掉网格曲面的隐藏线。

hidden off　命令显示网格曲面的隐藏线。

例如,下述示例程序 EX305 可绘制出图 3 - 8 所示的函数 $f(x,y)=\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$ 的网格曲面,其中左边的图形无隐藏线,而右边的图形含有隐藏线。

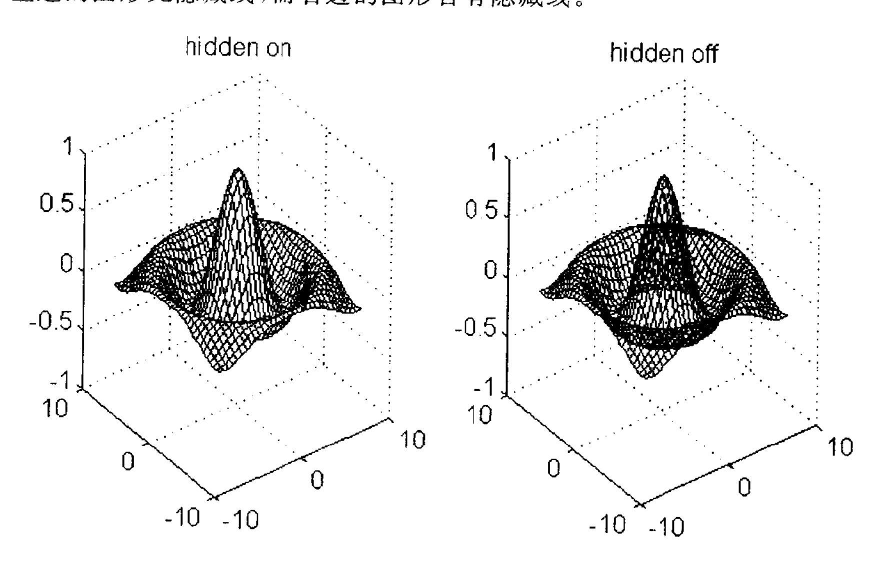


图 3 - 8　无隐藏线的网格曲面与有隐藏线的网格曲面

```
% ****************************
% 程序:EX305.M
% 功能:网格曲面的隐藏线
% ****************************
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X. ^ 2 + Y. ^ 2 ) + eps;
Z = sin(R). /R;
subplot( 1 , 2 , 1 )
mesh( X , Y , Z )
hidden on
grid on
title( ' hidden on ')
axis([ -10  10  -10  10  -1  1 ])
subplot( 1 , 2 , 2 )
mesh( X , Y , Z )
hidden off
grid on
title( ' hidden off ')
axis([ -10  10  -10  10  -1  1 ])
```

3.3 三维阴影曲面的绘制

前一节我们所绘制的三维曲面中,各个小的曲面片是由四边形组成的,这个四边形的 4 条边绘制有某一种颜色,但其内部却无颜色(即为绘图窗口的底色)。本节我们介绍另外一种三维曲面表示方法 —— 三维阴影曲面。这种曲面也是由很多个较小的四边形构成的,但各个四边形的 4 条边是无色的(即为绘图窗口的底色),其内部表面却分布有不同的颜色,也可认为是各个四边形带有阴影效果。MATLAB 提供了 3 条用于绘制这种三类阴影曲面的命令:surf,surfc,surfl。

3.3.1 阴影曲面绘制命令 surf

· 格式:

```
surf( X , Y , Z , C )
surf( X , Y , Z )
surf( x , y , Z , C )
surf( x , y , Z )
surf( Z , C )
surf( Z )
```

· 功能:

绘制 1 个三维阴影曲面。

说明：

① 这 6 个 surf 命令与 3.2.2 节所介绍的 6 个 mesh 命令的使用方法及参数含义相同。

② surf 命令与 mesh 命令的区别是前者绘制的是三维阴影曲面，而后者绘制的是三维网格曲面。

③ 在 surf 命令中，各个四边形表面的颜色分布方式可由 shading 命令来指定：

shading faceted —— 表示截面式颜色分布方式；

shading interp —— 表示插补式颜色分布方式；

shading flat —— 表示平面式颜色分布方式。

例如，下述示例程序 EX306 可绘制出图 3－9、图 3－10、图 3－11 所示的阴影曲面，这个阴影曲面是下述函数的图形

$$f(x,y)=\frac{2\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$$

其中，图 3－9、图 3－10、图 3－11 分别为设置了 shading faceted，shading interp 和 shading flat 的实际显示结果。

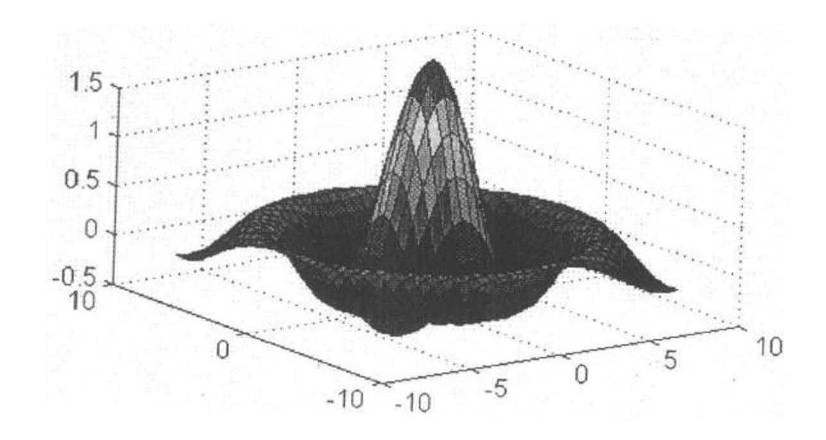


图 3－9　阴影曲面示例之一(shading faceted)

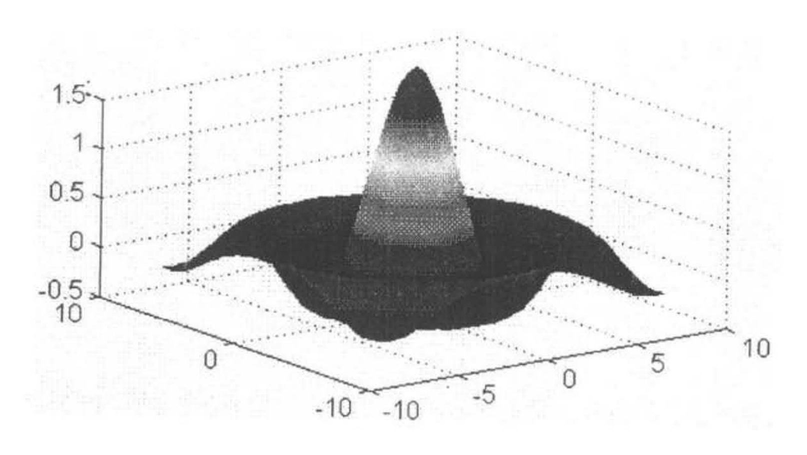


图 3－10　阴影曲面示例之二(shading interp)

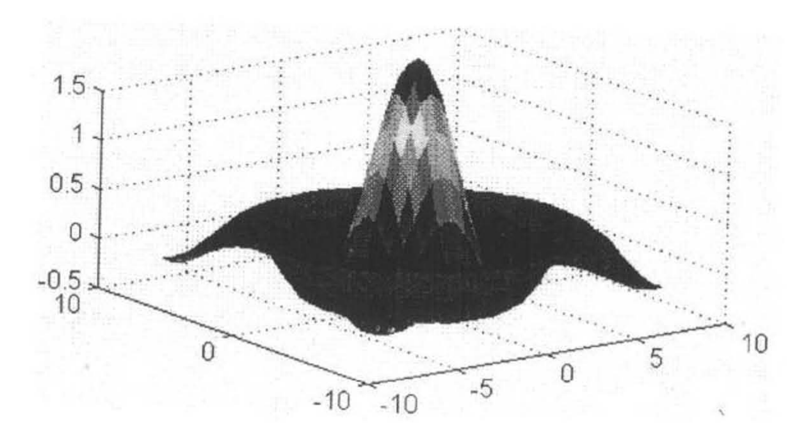


图 3-11 阴影曲面示例之三(shading flat)

```
% ******************************
% 程序:EX306.M
% 功能:阴影曲面的绘制
% ******************************
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X.^2 + Y.^2 ) + eps;
Z = 2*sin(R)./R;
surf( X , Y , Z )
grid on
axis([ -10 10 -10 10 -0.5 1.5 ])
shading faceted
```

3.3.2 带有等高线的阴影曲面绘制命令 surfc

- 格式:

```
surfc( X , Y , Z , C )
surfc( X , Y , Z )
surfc( x , y , Z , C )
surfc( x , y , Z )
surfc( Z , C )
surfc( Z )
```

- 功能:

绘制在 XY 平面上带有等高线的三维阴影曲面。

- 说明:

① 这 6 个 surfc 命令与 3.2.3 小节所介绍的相应的 6 个 meshc 命令的使用方法及参数含义相同。

② surfc 命令与 surf 命令的区别是前者除了绘制出三维阴影曲面外,在 XY 坐标平面上

还绘制有曲面在Z轴方向上的等高线，而后者仅绘制出三维阴影曲面。

例如，图3-12所示为带有函数等高线的阴影曲面，它是由surfc命令绘制出来的。

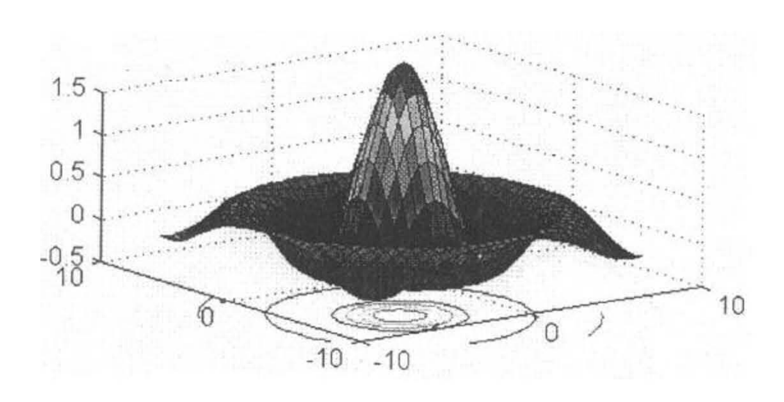


图3-12 带有函数等高线的阴影曲面

3.3.3 具有光照效果的阴影曲面绘制命令surfl

· 格式：

```
surfl ( X , Y , Z , s )
surfl ( X , Y , Z )
surfl ( Z , s )
surfl ( Z )
```

· 功能：

绘制具有光照效果的三维阴影曲面。

· 说明：

① 这4个surfl命令与前面介绍的surf命令的使用方法及参数含义相类似。

② surfl命令与surf命令的区别是前者绘制出的三维阴影曲面具有光照效果，而后者绘制出的三维阴影曲面无光照效果。

③ 向量参数s表示光源的坐标位置，s = [sx , sy , sz]。注意，若缺省s，则表示光源位置设在观测角的反时针45°处，它是缺省的光源位置。

例如，下述示例程序EX307可绘制出图3-13所示的具有光照效果的阴影曲面。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX307.M
% 功能:具有光照效果的阴影曲面的绘制
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X. ^ 2 + Y. ^ 2 ) + eps;
Z = 2 * sin(R). /R;
s = [ 0  -1  0 ];
```

```
surfl( X , Y , Z , s )
grid on
axis([ -10  10  -10  10  -0.5  1.5 ])
```

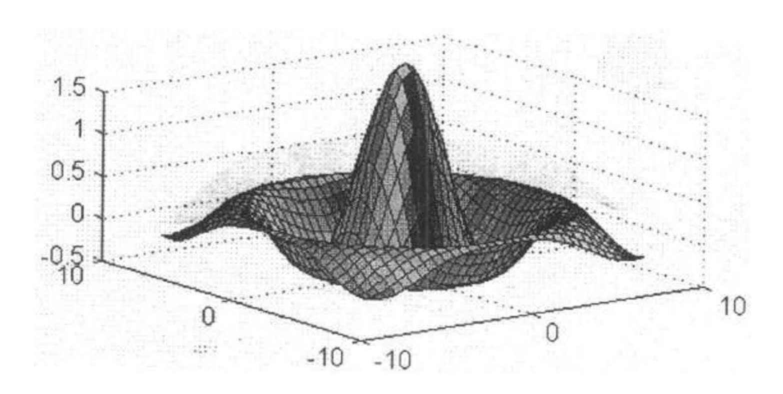


图 3-13 具有光照效果的阴影曲面示例

3.4 三维图形的调控

与二维图形一样,我们也可对三维图形的显示参数进行更改,以控制其显示效果。这里我们主要介绍设置视点位置的命令和控制坐标轴范围的命令。

3.4.1 设置视点位置

在纸上画出的或在屏幕上显示出的三维图形,其实质是在二维的载体上体现了三维的显示效果。在生活中,我们可能都有这样的经验,对同一个三维物体,当我们在不同的角度对其进行观察时,会有不同的视觉效果。在屏幕上显示三维图形时,也应如此,即取不同的视点,会得到不同的图形。

视点可以由两个方式来表示。一种方式是指定视点在三维直角坐标系中的坐标值(x , y , z),另一种方式是指定视线的方向,它可由向量

[az , el]

来表示。向量元素 az 和 el 的含义如图 3-14 所示,其中参数 az 表示在 XY 平面上,观测点与坐标原点连线的投影与-Y 轴的夹角大小(右手坐标系);参数 el 表示观测点与坐标原点的连线与 XY 平面的夹角大小(el 取正值时,表示观测点位于 XY 平面之上,el 取负值时,表示观测点位于 XY 平面之下)。

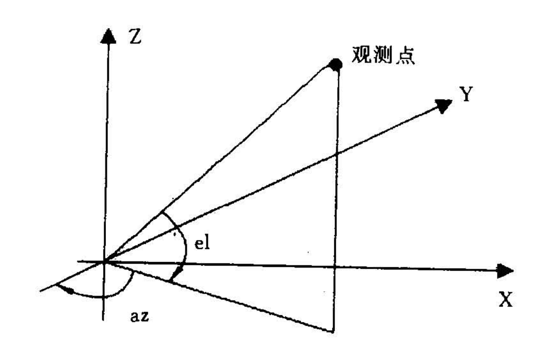


图 3-14 观测点的定义

MATLAB 提供了设定视点位置的命令 view。其使用方法如下:

- 格式：

```
view( [ x , y , z ] )
view( [ az , el ] ) 或 view( az , el )
view(2)
view(3)
[ x , y ] = view
```

- 功能：

① 命令 view([x , y , z])指定视点的位置是(x , y , z)。

② 命令 view([az , el]) 或 view(az , el)指定视点的定义角度。

③ 命令 view(2)设定[az , el] = [0 ,90]，即观测 XY 平面的显示效果。

④ 命令 view(3)设定[az , el] = [−37.5 ,30]，该观测值为 MATLAB 的缺省视点位置。

⑤ 命令[az , el] = view 返回当前视点的位置。

例如，下述示例程序 EX308 可绘制出图 3 - 15 所示图形，在该图形中对函数 $f(x,y)=\frac{2\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$设置了不同的视点进行观测。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX308.M
% 功能:具有不同观测视点的网格曲面
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
x = -8 : 0.5 : 8;
y = x;
[ X , Y ] = meshgrid( x , y );
R = sqrt( X. ^ 2 + Y. ^ 2 ) + eps;
Z = 2 * sin(R). /R;
subplot(2,2,1)
mesh( X , Y , Z )
view([0,0])
title('view( [ 0 , 0 ] )')
xlabel('X')
ylabel('Y')
zlabel('Z')
grid on
subplot(2,2,2)
mesh( X , Y , Z )
view([90,0])
title('view( [90 , 0 ] )')
xlabel('X')
ylabel('Y')
zlabel('Z')
grid on
subplot(2,2,3)
```

```
mesh( X , Y , Z )
view(3)
title('view( 3 )')
xlabel('X')
ylabel('Y')
zlabel('Z')
grid on
subplot(2,2,4)
mesh( X , Y , Z )
view(2)
title('view( 2 )')
xlabel('X')
ylabel('Y')
zlabel('Z')
grid on
```

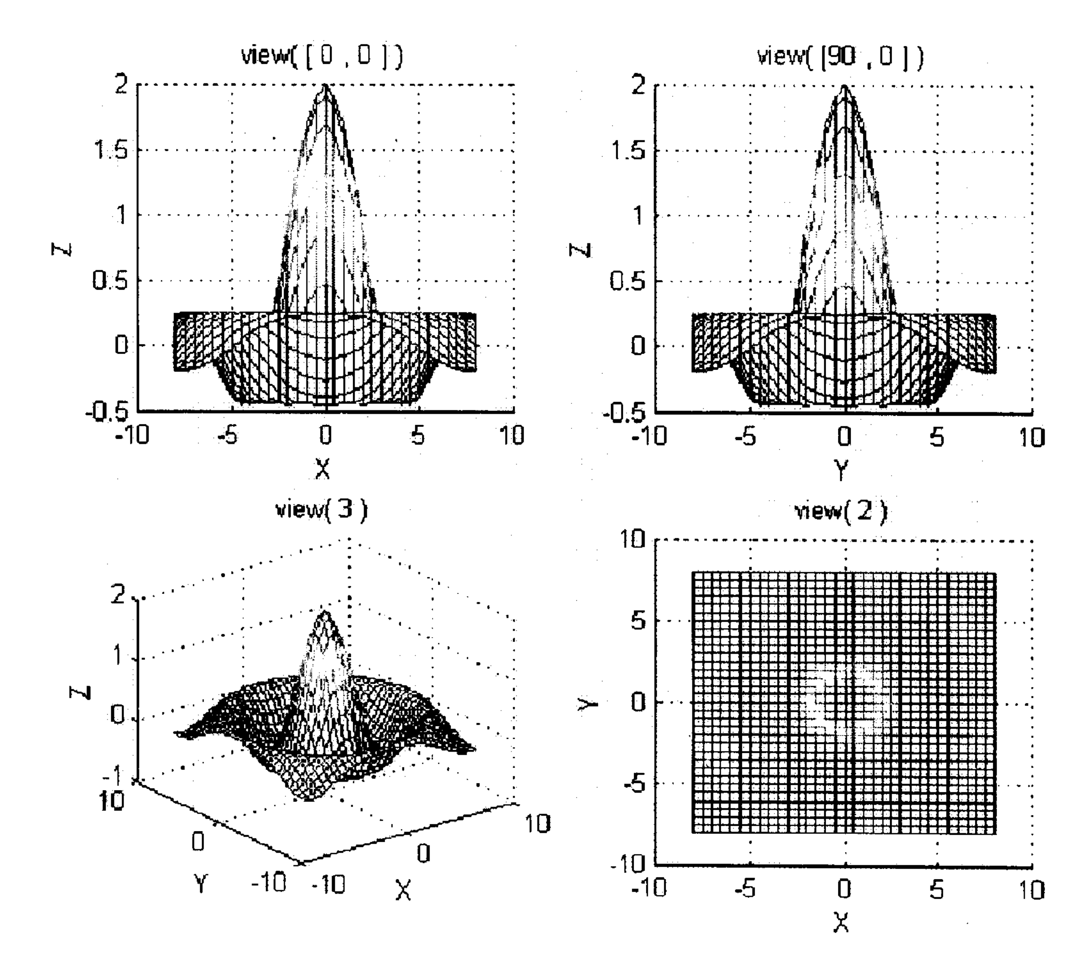


图 3 - 15　不同观测视点下的网格曲面

3.4.2 调整坐标轴的范围

与二维图形相类似，我们也可使用 axis 命令来调整三维图形各个坐标轴的显示范围。

· 格式：

axis([xmin xmax ymin ymax zmin zmax])

· 功能：

将所画图形的 X 轴的大小范围限定在{ xmin , xmax }之间，Y 轴的大小范围限定在{ ymin , ymax }之间，Z 轴的大小范围限定在{ zmin , zmax }之间。

· 说明：

二维图形中所用到的 axis 命令的其它一些格式也可应用于三维图形的绘制中，但要注意的是个别命令的操作有些细微的差别。

3.5 特殊三维图形的绘制

在科学研究中，我们有时也需要绘制一些特殊的三维图形，如统计学中的三维直方图、三维流场等。MATLAB 也提供了绘制这些实用三维图形的命令。

3.5.1 三维直方图的绘制

与二维情况相类似，MATLAB 提供了两类画三维直方图的命令：一类是用于画垂直放置的三维直方图，另一类是用于画水平放置的三维直方图。

一、画垂直放置的三维直方图

· 格式：

bar3(Z)

bar3(Y , Z)

bar3(Z , option)

· 功能：

① 命令格式 bar3(Z)以 x = 1,2,3,…,m 为各个数据点的 x 坐标，以 y = 1,2,3,…,n 为各个数据点的 y 坐标，以 ***Z*** 矩阵的各个对应元素为 z 坐标(***Z*** 矩阵的维数为 m×n)，画出一个垂直放置的三维直方图。

② 命令格式 bar3(Y , Z)以 x = 1,2,3,…,m 为各个数据点的 x 坐标，以 ***Y*** 向量的各个元素为各个数据点的 y 坐标，以 ***Z*** 矩阵的各个对应元素为 z 坐标(***Z*** 矩阵的维数为 m×n)，画出一个垂直放置的三维直方图。

③ 命令格式 bar3(Z , option)以 x = 1,2,3,…,m 为各个数据点的 x 坐标，以 y = 1,2,3,…,n 为各个数据点的 y 坐标，以 ***Z*** 矩阵的各个对应元素为 z 坐标(***Z*** 矩阵的维数为 m×n)，画出一个垂直放置的三维直方图，并且各个方块的放置位置由字符串参数 option 按下述方式来指定：

detached —— 分离式的三维直方图；

grouped —— 分组式的三维直方图；

stacked —— 累加式的三维直方图。

二、画水平放置的三维直方图

· 格式:

```
bar3h( Z )
bar3h( Y , Z )
bar3h( Z , option )
```

· 功能:

这 3 个命令的功能及使用方法与前述 3 个 bar3 命令的功能及使用方法相同,它们的区别在于所画出的直方图是水平放置的,而不是垂直放置的。

例如,下述示例程序 EX309 可绘制出图 3 - 16 所示的几个不同类型的直方图。

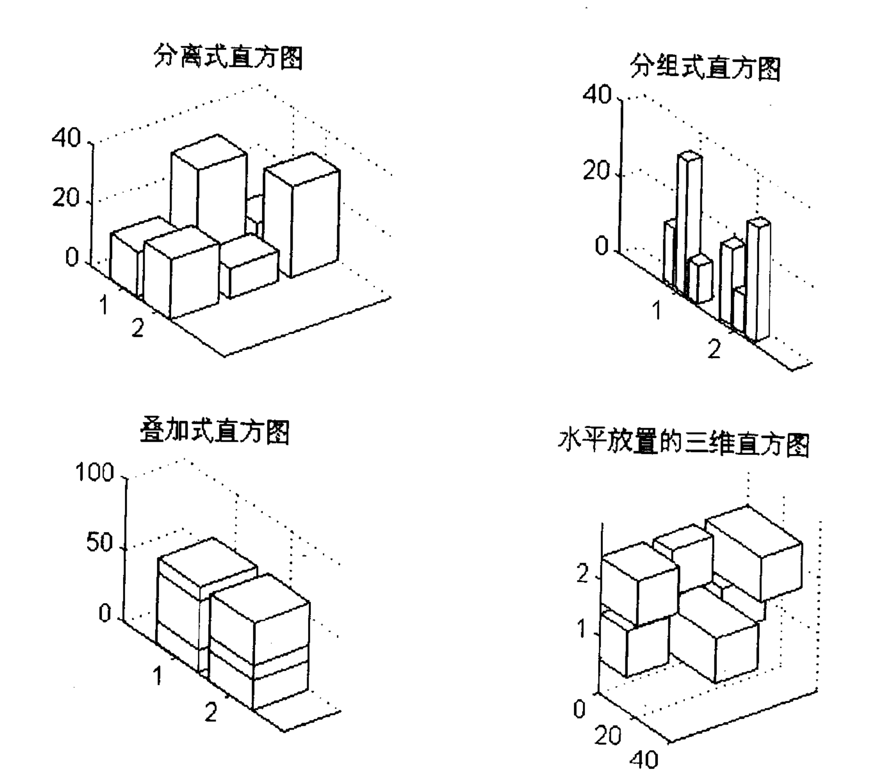


图 3 - 16　各种三维直方图绘制示例

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX309.M
% 功能:不同类型三维直方图绘制示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Z = [ 15 , 35 , 10 ; 20 , 10 , 30 ];
subplot( 2 , 2 , 1 )
h1 = bar3( Z , 'detached' )
set( h1 , 'FaceColor' , 'W' )
title( '分离式直方图' )
subplot( 2 , 2 , 2 )
```

```
h2 = bar3( Z , 'grouped' )
set( h2 , 'FaceColor' , 'W' )
title( '分组式直方图' )
subplot( 2 , 2 , 3 )
h3 = bar3( Z , 'stacked' )
set( h3 , 'FaceColor' , 'W' )
title( '叠加式直方图' )
subplot( 2 , 2 , 4 )
h4 = bar3h( Z )
set( h4 , 'FaceColor' , 'W')
title( '水平放置的三维直方图' )
```

3.5.2 三维离散序列图的绘制

MATLAB 也提供了绘制三维离散序列图的命令。

- 格式：

```
stem3( Z )
stem3( X , Y , Z , option )
stem3( X , Y , Z , 'filled' )
```

- 功能：

① 命令 stem3(Z)以 x = 1,2,3,…,m 为各个数据点的 x 坐标,以 y = 1,2,3,…,n 为各个数据点的 y 坐标,以 **Z** 矩阵的各个对应元素为 z 坐标(**Z** 矩阵的维数为 m×n),在(x , y , z)坐标点画 1 个空心的小圆圈,并连接一条线段到 XY 坐标轴。

② stem3(X , Y , Z , option)以向量 ***X*** 的各个元素为 x 坐标,以向量 ***Y*** 的各个元素为 y 坐标,以 **Z** 矩阵的各个对应元素为 z 坐标,在(x , y , z)坐标点画 1 个空心的小圆圈,并连接一条线段到 X 坐标轴。

③ stem3(X , Y , Z , 'filled')以向量 ***X*** 的各个元素为 x 坐标,以向量 ***Y*** 的各个元素为 y 坐标,以 **Z** 矩阵的各个对应元素为 z 坐标,在(x , y , z)坐标点画 1 个实心的小圆圈,并连接一条线段到 XY 坐标轴。

- 说明：

① 字符串参数 option 是 1 个可选参数,它表示绘制图形时的线型、颜色,其取值见表 2 - 1。

② 小圆圈的填充颜色也是由参数 option 来说明的。

例如,下述示例程序 EX310 可绘制出图 3 - 17 所示的离散序列图。

```
% ********************************
% 程序:EX310.M
% 功能:三维离散序列图绘制示例
% ********************************
t = 0 : pi/10 : 6 * pi;
x = exp(-t/10). * cos(t);
y = 2 * exp(-t/10). * sin(t);
stem3( x , y , t , 'filled' )
```

```
hold on
plot3( x , y , t )
xlabel('X')
ylabel('Y')
zlabel('t')
```

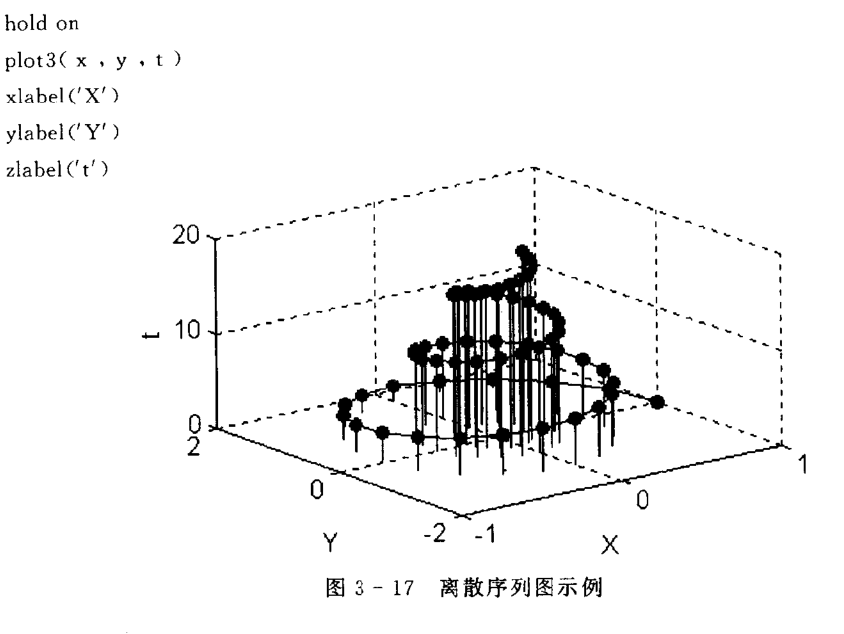


图 3 - 17 离散序列图示例

3.5.3 球面的绘制

MATLAB 提供的绘制球面的方法是:先利用球面的表面数据产生命令 sphere 产生一组单位球面的坐标矩阵值,然后由该矩阵值利用前面所介绍的 mesh 命令或 surf 命令来绘制出球面图形。

· 格式

```
sphere( n )
[ X , Y , Z ] = sphere( n )
```

· 功能

① 命令 sphere(n)产生单位球面上的数据点,并直接由 surf 命令绘制出这个单位球面。

② 命令[X , Y , Z] = sphere(n)产生 3 个维数为(n+1)×(n+1)的矩阵 ***X***,***Y***,***Z***(不直接绘制出球面图形),它们分别表示单位球面上的一系列数据点 (x , y , z) 的坐标值。利用这些矩阵数据,可再由 mesh 命令或 surf 命令来绘制出指定大小和位置的球面图形。

· 说明

参数 n 确定了球面绘制的精度。n 值越大,则数据点越多,绘制出的球面就越精确。反之,n 值越小,精度越低。n 的缺省值是 n = 20。

例如,下述示例程序 EX311 可绘制出图 3 - 18 所示的两种球面。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX311.M
% 功能:球面绘制示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
subplot( 1 , 2 , 1 )
sphere( 25 );
title('单位球面')
```

```
subplot( 1 , 2 , 2 )
[X,Y,Z] = sphere(25);
mesh( X , Y , 2 * (Z+1) )
title('移位和放大了的球面')
```

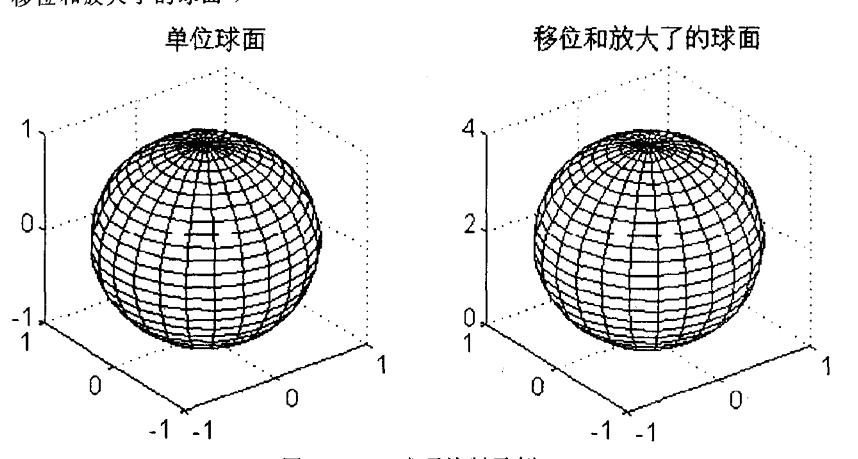


图 3 - 18　球面绘制示例

3.5.4　圆柱体的绘制

MATLAB 提供的绘制圆柱体的方法是:先利用圆柱体表面数据产生命令 cylinder 产生一组圆柱体表面的坐标矩阵值,然后由该矩阵值利用前面所介绍的 mesh 命令或 surf 命令来绘制出这个圆柱体图形。注意,圆柱体是由其基线 r 绕 Z 轴旋转一周而成的,所以绘制圆柱体之前必须先知道其基线 r。这里的基线是由其各个点与 Z 轴的距离(即半径)来定义的。

· 格式

```
[ X , Y , Z ] = cylinder( r , n )
cylinder( r , n )
cylinder( r )
cylinder
```

· 功能

绘制一个圆柱体。

· 说明

① 命令[X , Y , Z] = cylinder(r , n)产生 3 个维数为(n+1)×(n+1)的矩阵 ***X***,***Y***,***Z***(不直接绘制出圆柱体),它们分别表示圆柱体表面上一系列数据点 (x , y , z) 的坐标值。利用这些矩阵数据,可再由 mesh 命令或 surf 命令来绘制出指定大小和位置的圆柱体图形。

② 参数 r 为 1 个向量,它表示等距离分布的沿圆柱体基线在其单位高度的半径。***r*** 的缺省值是 ***r*** = [1　1]。

③ 参数 n 确定了圆柱体绘制的精度。n 值越大,则数据点越多,绘制出的圆柱体就越精确。反之,n 值越小,精度越低。n 的缺省值是 n = 20。

④ 命令 cylinder(r , n)产生圆柱体表面上的数据点,并直接由 surf 命令绘制出这个圆柱体。

⑤ 命令 cylinder(r)以缺省的参数 n=20 绘制基线为 r 的圆柱体。

⑥ 命令 cylinder 以缺省的参数 $r = [1 \quad 1]$，n = 20 绘制单位圆柱体。例如，下述示例程序 EX312 可绘制出图 3－19 所示的两种圆柱体。

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * *
% 程序:EX312.M
% 功能:圆柱体绘制示例
% * * * * * * * * * * * * * * * * * * * * * * * * * * *
subplot( 1 , 2 , 1 )
[X,Y,Z] = cylinder;
mesh( X , Y , Z )
title('单位圆柱体')

subplot( 1 , 2 , 2 )
t=1:10;
r(t)=t. * t;
[X,Y,Z] = cylinder(r,40);
mesh( X , Y , Z )
title('一般圆柱体')
```

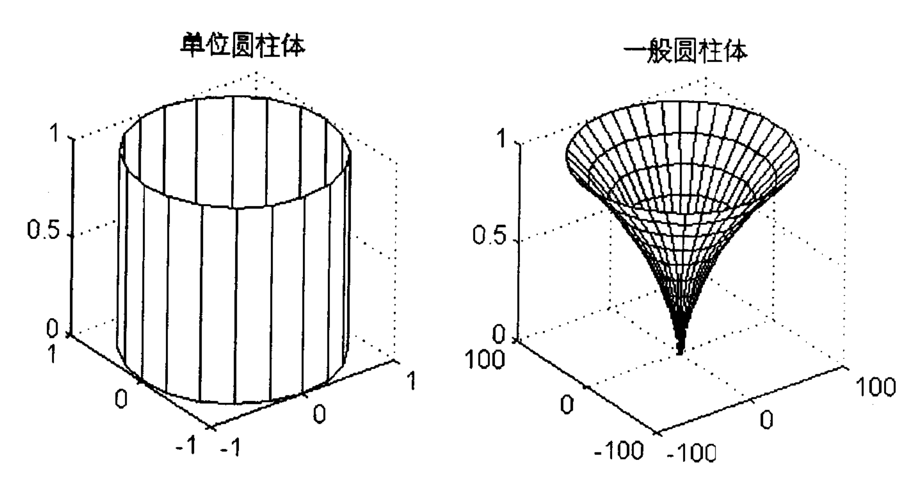


图 3－19　圆柱体绘制示例

3.5.5 等高线的绘制

- 格式

```
contour( Z , option )
contour( x , y , Z , option )
contour( Z , n , option )
contour( x , y , Z , n , option )
contour3( Z , option )
contour3( X , Y , Z , option )
contour3( Z , n , option )
```

```
contour3( X , Y , Z , n , option )
clabel( c , h )
```

· 功能

① contour 命令绘制二维的等高线。

② contour3 命令绘制三维的等高线。

③ clabel 命令标记等高线的数值。

· 说明

① 参数 n 指定要绘制出 n 条等高线。若缺省参数 n,则系统自动确定绘制等高线的条数。

② 若指定向量参数 x 和 y,则以 x 向量值为 X 轴,y 向量值为 Y 轴,绘制二维等高线。

③ 若指定矩阵参数 X 和 Y,则基于矩阵 X 和 Y 绘制三维等高线。

④ 选项参数 option 是一个字符串参数,它指定了等高线的线型和颜色,其指定方法见表 2-1。也可省略该参数,此时表明 option='—y',即使用黄色的实线。

⑤ clabel 命令中的参数(c,h)必须是 contour 命令的返回值,即

```
[ c , h ] = contour( …… )
```

例如,下述示例程序 EX313 可绘制出图 3-20 所示的曲面及其对应的二维等高线。该曲面的方程是 Peaks 函数

$$f(x,y)=3(1-x)^2e^{-x^2-(y+1)^2}-10(\frac{x}{5}-x^3-y^5)e^{-(x^2-y^2)}-\frac{1}{3}e^{-(x+1)^2-y^2}$$

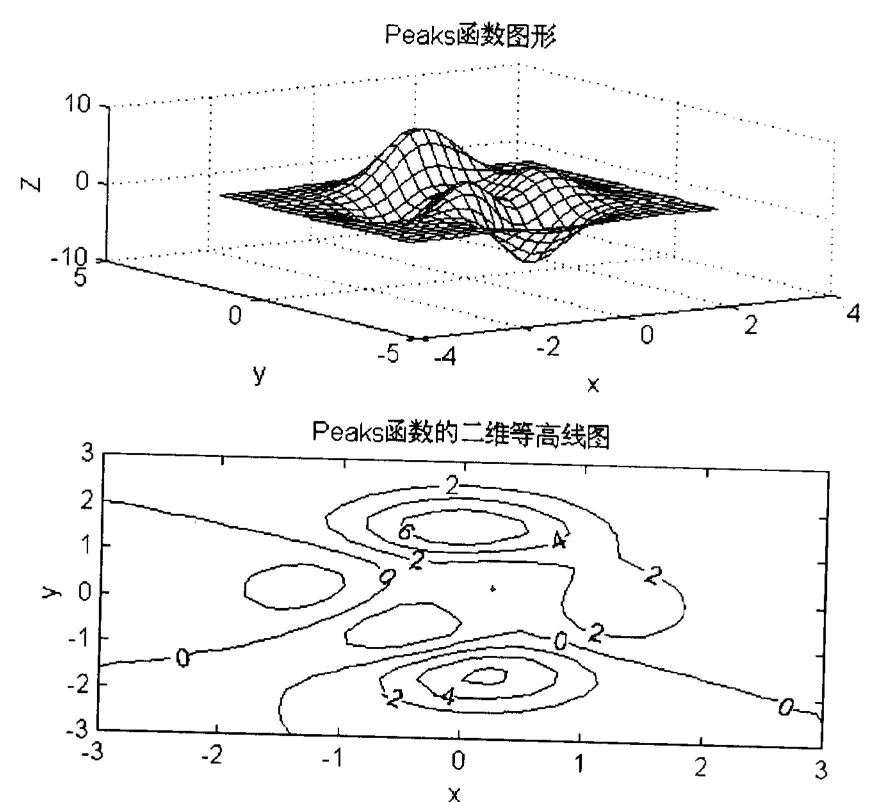


图 3-20 二维等高线绘制示例

```
% ****************************
% 程序:EX313.M
% 功能:曲面及其对应的二维等高线绘制示例
% ****************************
x = -3:0.25:3;
y = x;
[X,Y] = meshgrid(x,y);
Z =  3*(1-X).^2.*exp(-(X.^2) - (Y+1).^2)...
    - 10*(X/5 - X.^3 - Y.^5).*exp(-X.^2-Y.^2)...
    - 1/3*exp(-(X+1).^2 - Y.^2);

subplot(2,1,1)
mesh(X,Y,Z)
xlabel('x')
ylabel('y')
zlabel('Z')
title('Peaks 函数图形')

subplot(2,1,2)
[c,h] = contour(x,y,Z);
clabel(c,h)
xlabel('x')
ylabel('y')
zlabel('Z')
title('Peaks 函数的二维等高线图')
```

又例如,下述示例程序 EX314 可绘制出图 3-21 所示的曲面及其对应的三维等高线。

```
% ***************************
% 程序:EX314.M
% 功能:曲面及其对应的三维等高线绘制示例
% ***************************
x = -3:0.25:3;
y = x;
[X,Y] = meshgrid(x,y);
Z =3*(1-X).^2.*exp(-(X.^2) - (Y+1).^2)...
    - 10*(X/5 - X.^3 - Y.^5).*exp(-X.^2-Y.^2)...
    - 1/3*exp(-(X+1).^2 - Y.^2);

subplot(2,1,1)
mesh(X,Y,Z)
xlabel('x')
ylabel('y')
zlabel('Z')
title('Peaks 函数图形')
```

```
subplot(2,1,2)
[c,h] = contour3(x,y,Z);
clabel(c,h)
xlabel('x')
ylabel('y')
zlabel('Z')
title('Peaks 函数的三维等高线图')
```

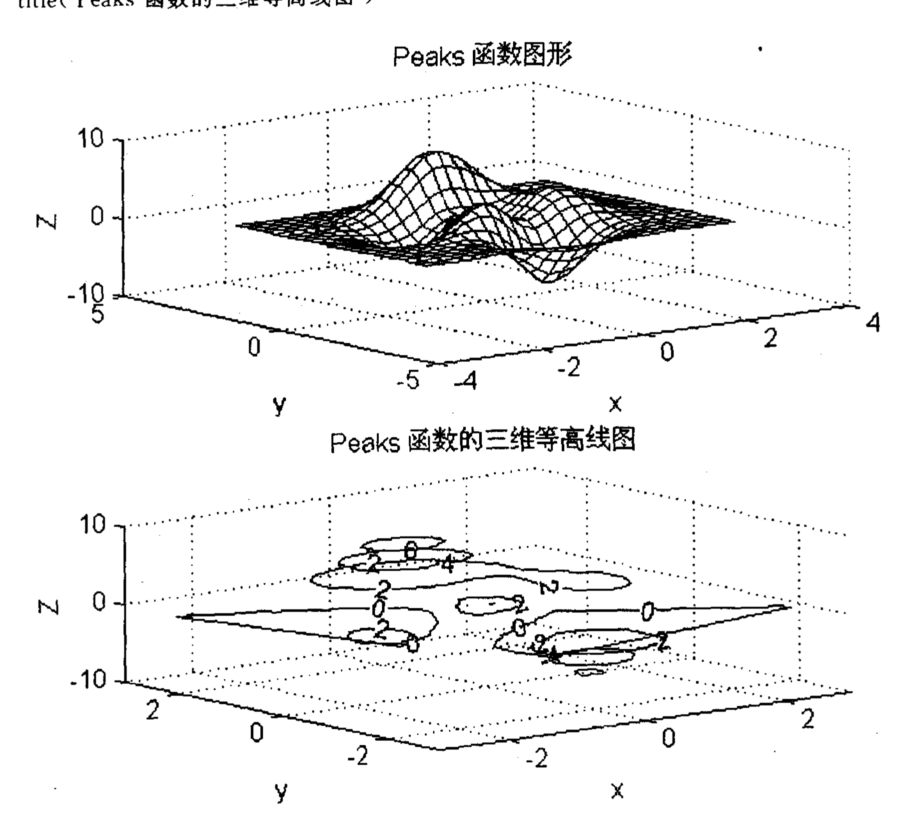


图 3-21　三维等高线绘制示例

3.5.6　梯度场的绘制

在工程计算中，梯度场常用一系列方向、大小不同的箭头来表示。MATLAB 提供了绘制二维和三维梯度场的命令。

- 格式

quiver(X , Y , DX , DY , scale , option)

quiver3(X , Y , Z , DX , DY , DZ , scale , option)

- 功能

① 命令 quiver 绘制二维梯度场。

② 命令 quiver3 绘制三维梯度场。

· 说明

① 参数 X、Y、Z 定义了用于表示梯度场的箭头的所在位置的坐标值。

② 参数 DX、DY、DZ 定义了箭头的大小和方向。一般，它们可由梯度函数 graduent 求得

$$DX = gradient(x)$$

$$DY = gradient(y)$$

$$DZ = gradient(z)$$

③ 参数 scale 用于控制是否放大或缩小箭头的长度。

④ 选项参数 option 是一个字符串参数，它指定了等高线的线型和颜色，其指定方法见表 2 - 1。也可省略该参数，此时表明 option='—y'，即使用黄色的实线。

例如，下述示例程序 EX315 可绘制出图 3 - 22 所示的二维梯度场。

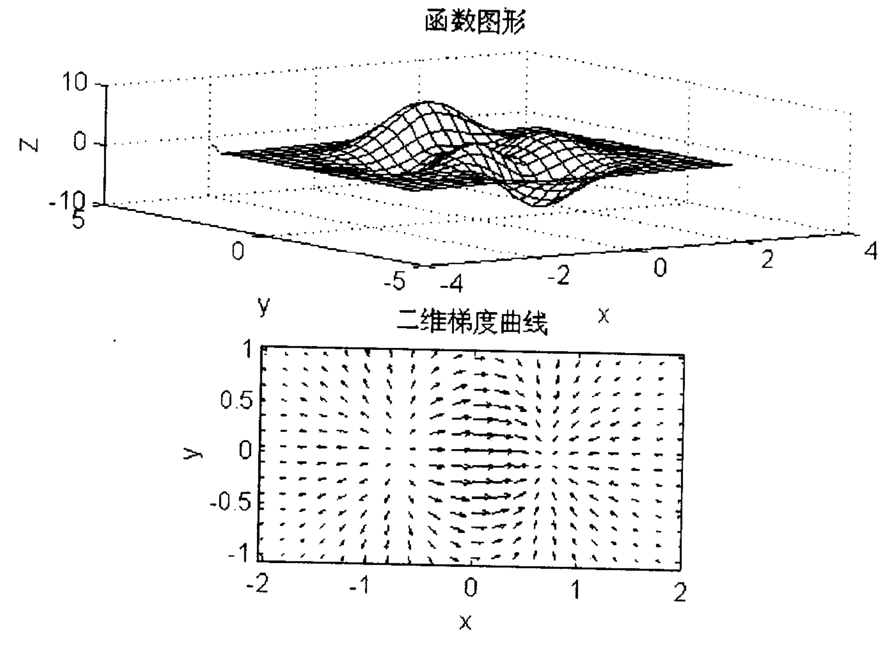


图 3 - 22　二维梯度场绘制示例

```
% ***************************
% 程序:EX315.M
% 功能:二维梯度场绘制示例
% ***************************
[ x , y ] = meshgrid(-2:0.2:2,-1:0.15:1);
z = x. * exp(-x. ^ 2-y. ^ 2);
[dx,dy] = gradient(z,0.2,0.15);

subplot(2,1,1)
mesh(X,Y,Z)
xlabel('x')
ylabel('y')
```

```
zlabel('Z')
title('函数图形')
hold on

subplot(2,1,2)
contour(x,y,z)
quiver(x,y,dx,dy),
axis image
xlabel('x')
ylabel('y')
title('二维梯度曲线')
hold off
```

又例如，下述示例程序 EX316 可绘制出图 3－23 所示的三维梯度场。

```
% ****************************
% 程序:EX316.M
% 功能:三维梯度场绘制示例
% ****************************
[x,y] = meshgrid( -2:0.3:2 , -1:0.2:1 );
z = x. * exp(-x. ^ 2-y. ^ 2);
[u,v,w] = surfnorm(x,y,z);
quiver3(x,y,z,u,v,w);
hold on
mesh(x,y,z)
xlabel('x')
ylabel('y')
zlabel('Z')
hold off
```


图 3－23　三维梯度场绘制示例

第4章 MATLAB的图形对象及其属性

MATLAB采用了矢量化的绘图方法。与位图(bitmap)的方式不一样,矢量化绘图方法所绘制出的图形中,各个图形元素是相互独立的,可单独地进行修改处理。这种独立的图形元素称为图形对象。MATLAB总共提供了10种不同类型的基本图形对象,本章就介绍这些基本图形对象的属性及其修改方法,具体包括下述内容:

- MATLAB的图形对象及其结构关系
- 图形对象属性值的获取与设置方法
- MATLAB各种图形对象的属性详解

4.1 MATLAB的图形对象及其结构关系

4.1.1 MATLAB的图形对象

我们在MATLAB中所绘制出的图形是由一些称为图形对象的基本图形元素所构成的。不管我们用哪种命令绘制出的图形,它都是由一些图形对象组成的。各个图形对象的具体显示效果,是依据其所具有的能控制其特征或性能的属性值来确定的。

MATLAB共定义了10个不同的图形对象:Root(根对象),Figure(图形窗口对象),Axes(坐标轴对象),Line(直线对象),Patch(多边形区域对象),Surface(曲面对象),Image(图像对象),Text(文字对象),Uimenu(菜单对象),Uicontrol(控件对象)。这些图形对象构成了MATLAB中各种图形的基础。下面我们对这些基本图形对象进行简要说明。

一、Root对象

英文中root的含义是指"根,树根"。顾名思义,Root对象表示的就是整个图形的"根"。计算机中图形的显示载体是屏幕,屏幕成了全部图形的"根",所以MATLAB中的Root对象指的就是计算机屏幕。注意,系统中只有一个Root对象。

二、Figure对象

Figure对象是指在屏幕上所产生的一个一个的图形窗口。MATLAB允许用户使用Figure指令生成任意数目的图形窗口,每个图形窗口就构成了一个Figure对象。

三、Axes对象

Axes对象是指在图形窗口中所设置的一个坐标轴。坐标轴控制了绘图区域的大小,MATLAB所绘制的直线、曲线、曲面等各种图形,都不能超过Axes对象所指定的绘图范围。Axes对象可由axis指令来生成。另外,在一些高级绘图命令中(如plot,plot3,contour,mesh,surf等命令),都包括有自动生成Axes对象的命令。

四、Line 对象

Line 对象是指在 Axes 对象所确定的绘图范围内的一条折线，它是构成二维图形及三维图形的最基本的绘图元素。Line 对象可由 line 命令产生。另外，在一些高级绘图命令中(如 plot，plot3 等命令)，都包括有自动生成 Line 对象的命令。

五、Patch 对象

Patch 对象是指在 Axes 对象所确定的绘图范围内的指一个填充了的多边形。Patch 对象可由 fill，fill3 命令来生成。

六、Surface 对象

Surface 对象是指在 Axes 对象所确定的绘图范围内的一个以矩阵数据所表示的三维曲面图形，这个曲面图形是由一个一个的较小的四边形相互连接在一起所形成的。Surface 对象可由 surf，mesh 等命令产生。

七、Image 对象

Image 对象是指在 Axes 对象所确定的绘图范围内的 1 幅图像。Image 对象可由 image 指令产生。

八、Text 对象

Text 对象是指图形窗口中的一串文字。Text 对象可由 text 指令生成。另外，xlabel，ylabel，zlabel，clabel，title 等一些设置字符串的命令都包含有自动生成 Text 对象的命令。

九、Uimenu 对象

Uimenu 对象是指在图形窗口中所产生的 1 个下拉式菜单。每个图形窗口中一般只有 1 个 Uimenu 对象。

十、Uicontrol 对象

Uicontrol 对象是指在图形窗口中所产生的 1 种控制部件，它引导程序去完成某种功能或实现程序与用户之间的信息交换。

4.1.2 图形对象之间的结构关系

MATLAB 所绘制出的图形都是由它所定义的 10 个基本的图形对象组合而成的。在 MATLAB 绘制图形的过程中，它所涉及到的一些图形对象将按一定的次序生成。

例如，当我们使用 plot 这个高级绘图命令绘制 1 个二维折线时，MATLAB 的执行过程大致如下：

(1) 首先使用 figure 命令，在屏幕(Root 对象)上生成 1 个图形窗口(Figure 对象)；

(2) 其次使用 axis 命令，在图形窗口内生成一个绘图区域(Axes 对象)；

(3) 最后再用 line 命令在 Axes 所指定的绘图区域内绘制所要求的折线(Line 对象)。

从上面这个绘图步骤可以看出，各个图形对象的产生过程是：Root→Figure→Axes→Line，我们可以把这个过程认为是一种层次关系。

MATLAB 所定义的 10 个图形对象之间的层次关系如图 4-1 所示。

这种层次关系是一种树状结构，其上一层是下一层的“双亲”，而下一层是上一层的“孩子”。如 Root 对象是 Figure 对象的“双亲”，而 Figure 对象又是 Uimenu，Uicontrol，Axes 对象的“双亲”，Line 对象又是 Axes 对象的“孩子”。

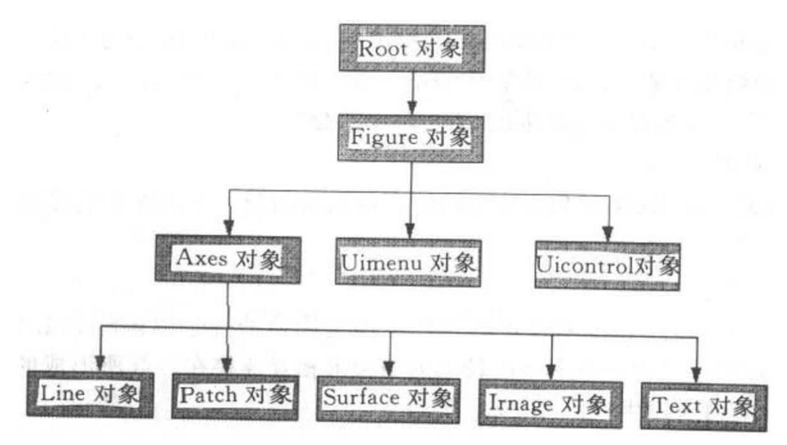


图 4-1　图形对象及其层次结构关系

4.1.3　图形对象的标识 —— 句柄

前面我们说过，任何一个 MATLAB 图形都是由一些基本的图形对象构成的，用户可以设定或更改这些图形对象的对外显示特性及特点。为确定具体是更改了哪一个图形对象的显示特性，在更改时就要指明其操作对象。MATLAB 给图形中的各个图形对象指定一个句柄(handle)，句柄惟一地标识出了我们要进行操作的图形对象。

MATLAB 中各个图形对象的句柄是由一些相关的命令在其执行过程中自动生成的。对于 MATLAB 的 10 个基本图形对象，它们各自的相应句柄的主要生成方法是：

(1) Root 对象的句柄是屏幕，这是 MATLAB 的规定，不用重新生成。MATLAB 还规定 Root 对象的句柄值为 0。

(2) Figure 对象的句柄生成指令是:

```
handle = figure('PropertyName', PropertyValue , … )
```

即可以直接设定其属性值。

另外，由于 Figure 对象是经常要用到的一个图形对象，所以 MATLAB 还提供了另外一个提取其句柄的常用函数:

```
handle = gcf
```

其作用是返回当前图形窗口的句柄到 handle 变量。

(3) Axes 对象的句柄生成指令是:

```
handle = axes('PropertyName' , PropertyValue , … )
```

即可以直接设定其属性值。

另外，由于 Axes 对象也是经常要用到的一个图形对象，所以 MATLAB 还提供了另外一个提取其句柄的常用函数:

```
handle = gca
```

其作用是返回当前坐标轴的句柄到 handle 变量。

(4) Line 对象的句柄生成指令是:

```
handle = line( x , y , z )
```

```
handle = line( 'PropertyName' , PropertyValue , … )
```

即可以直接设定其属性值。

(5) Patch 对象的句柄生成指令是:

```
handle = patch( x , y , c )
handle = patch( x , y , z , c )
handle = patch('XData' , x , 'YData' , y , 'ZData' , z , 'CData' , c )
handle = patch('PropertyName', PropertyValue , … )
```

即 patch 是一个低级绘图命令,它不仅可以绘制出填充多边形,取得其句柄值,而且也可以直接设定其属性值。

(6) Surface 对象的句柄生成指令是:

```
handle = surface( x , y , z )
handle = surface( x , y , z , c )
handle = surface( z , c )
handle = surface( z )
handle = surface('XData', x , 'YData' , y , 'ZData' , z , 'CData' , c )
handle = surface('PropertyName' , PropertyValue , … )
```

(7) Image 对象的句柄生成指令是:

```
handle = image( C )
handle = image( X , Y , C )
```

(8) Text 对象的句柄生成指令是:

```
handle = text( x , y , string )
handle = text( x , y , z , string )
handle = text('PropertyName', PropertyValue , … )
```

(9) Uimenu 对象的句柄生成指令是:

```
handle = uimenu( fighandle , 'PropertyName' , PropertyValue , … )
```

我们将在第 5 章对该命令作详细介绍。

(10) Uicontrol 对象的句柄生成指令是:

```
handle = uicontrol( fighandle , 'PropertyName', PropertyValue , … )
```

我们将在第 5 章对该命令作详细介绍。

4.2 图形对象属性值的获取与设定

从计算机软件编程的角度来讲,MATLAB 提供了面向对象的编程能力。对于每一个图形对象,我们不用了解其某些细节的实现方式,我们只须利用其各种属性(property)就能够控制该对象的外观特征及内在特点。

MATLAB 为其不同的图形对象提供了很多种控制其特征的属性。如 Figure 对象的 Color 属性可控制图形窗口的背景颜色,Axes 对象的 TickLength 属性可控制坐标轴刻度标记的长短,Text 对象的 Rotation 属性可控制文字串对象的显示方向,等等。

不同的图形对象有不同的属性,并且对各个属性的取值也有不同的限制。对于 MATLAB 所提供的这些属性,我们可通过 get 命令和 set 命令来提取及设定其属性值。

4.2.1 属性值的获取

- 格式：

PropertyValue = get(handle , 'PropertyName')

- 功能：

获取指定图形对象的某个指定属性的属性值。

- 说明：

① 对句柄为 handle 的图形对象，返回其 PropertyName 所指定属性的当前设定值，并将该属性值保存到变量 PropertyValue 中。该命令格式中的属性名必须用一对单引号括起来。

② 如果要返回某一属性的缺省值，必须在其属性名前加 'Default' 字样，即此时的命令格式为：

PropertyValue = get(handle ,'DefaultObjectTypePropertyName')

③ 如果要返回某一对象的所有属性值，则不需要指定其各个单独的属性名，即此时的命令格式为：

get(handle)

④ 如果要返回某一对象所有属性的缺省值，则可使用下列格式的 get 命令：

get(handle , 'Default')

例如，为了得到 Axes 对象的 Color 属性的缺省值，我们可以使用下述命令。

```
» get( gcf , 'DefaultAxesColor' )
ans =
        1         1         1
```

又例如，若使用 get(gcf) 命令则可以获得当前图形窗口的全部属性的当前所设定的属性值。

```
» get(gcf)
    BackingStore = on
    CloseRequestFcn = closereq
    Color = [0.8   0.8   0.8]
    Colormap = [ (64 by 3) double array]
    CurrentAxes = []
    CurrentCharacter =
    CurrentObject = [ ]
    CurrentPoint = [0 0]
    Dithermap = [ (64 by 3) double array]
    DithermapMode = manual
    FixedColors = [ (3 by 3) double array]
    IntegerHandle = on
    InvertHardcopy = on
    KeyPressFcn =
```

MenuBar = figure
MinColormap = [64]
Name =
NextPlot = add
NumberTitle = on
PaperUnits = inches
PaperOrientation = portrait
PaperPosition = [0.25   2.5   8   6]
PaperPositionMode = manual
PaperSize = [8.5   11]
PaperType = usletter
Pointer = arrow
PointerShapeCData = [(16 by 16) double array]
PointerShapeHotSpot = [1 1]
Position = [64 36 672 504]
Renderer = painters
RendererMode = auto
Resize = on
ResizeFcn =
SelectionType = normal
ShareColors = on
Units = pixels
WindowButtonDownFcn =
WindowButtonMotionFcn =
WindowButtonUpFcn =
WindowStyle = normal

ButtonDownFcn =
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
Interruptible = on
Parent = [0]
Selected = off
SelectionHighlight = on
Tag =
Type = figure
UserData = []
Visible = on

4.2.2 属性值的设定

· 格式:

set(handle , 'PropertyName1' , PropertyValue1 , 'PropertyName2', Property Value2 , …)

· 功能:

设定指定图形对象的某些指定属性的属性值。

· 说明:

① 对于句柄为 handle 的图形对象,上述 set 命令将其属性 PropertyName1 的属性值设定为 PropertyValue1,将属性 PropertyName2 的属性值设定为 PropertyValue2……即该命令可同时设定多个属性的属性值。

② 要显示某一对象的所有可设定的属性名称及其可能的取值时,可用下述命令:

set(handle)

③ 要显示某一对象的某一个属性的可能取值时,可用下述命令:

set(handle , 'PropertyName')

④ 要设定某一对象的某一个属性的缺省值时,可用下述命令:

set(handle ,'DefaultObjectTypePropertyName', PropertyValue)

例如,下面的命令先显示 Axes 对象的 Color 属性的缺省值,然后再更改这个属性的缺省值,最后再显示这个更改后的缺省值。

```
» get( gcf , 'DefaultAxesColor')
ans =
        1        1        1
» set(gcf , 'DefaultAxesColor' , [ 0.8   0.7   0.8 ] )
» get( gcf , 'DefaultAxesColor')
ans =
      0.8000     0.7000     0.8000
```

4.3 Root 对象的属性

1. AutomaticFileUpdates 属性

【描述】 AutomaticFileUpdates: [on | off]

【用途】 是否自动更新文件。

【说明】 on —— 自动更新文件;

off —— 不自动更新文件。

2. BusyAction 属性

【描述】 BusyAction: [{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队;

cancel —— 终止。

3. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

4. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的“子女”图形对象。

5. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

6. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

7. CurrentFigure 属性

【描述】 CurrentFigure：handle

【用途】 记忆当前图形窗口的句柄值。

8. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

9. Diary 属性

【描述】 Diary：[on | off]

【用途】 设置是否保存所有的执行命令及相应的执行结果。

【说明】 on —— 保存所有的执行命令及相应的执行结果；

off —— 不保存所有的执行命令及相应的执行结果。

10. DiaryFile 属性

【描述】 DiaryFile：filename

【用途】 定义 MATLAB 的“diary 文档”的文件名。

11. Echo 属性

【描述】 Echo：[on | off]

【用途】 设置是否回显 MATLAB 所执行的命令。

【说明】 on —— 回显 MATLAB 所执行的命令；

off —— 不回显 MATLAB 所执行的命令。

12. ErrorMessage 属性

【描述】 ErrorMessage：string

【用途】 错误信息描述。

13. Format 属性

【描述】 Format：[short|long|short E|long E|short G|long G|hex|bank|+|rational]

【用途】 设定 MATLAB 的输出格式。

【说明】 short —— 短格式；

long —— 长格式；

short E —— 短格式 E 方式；

long E —— 长格式 E 方式；

short G —— 短格式 G 方式；

long G —— 长短格式 G 方式；

hex —— 十六进制格式；

bank —— 银行格式；

+ —— 紧密格式；

rational —— 紧密格式。

14. FormatSpacing 属性

【描述】 FormatSpacing：[loose | compact]

【用途】 设置显示矩阵时是否压缩矩阵各行之间的换行符。

【说明】 loose —— 不压缩矩阵各行之间的换行符；

compact —— 压缩矩阵各行之间的换行符。

15. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

16. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

17. Language 属性

【描述】 Language：string

【用途】 说明所使用的语言种类。

18. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的"双亲"图形对象。

19. PointerLocation 属性

【描述】 PointerLocation：[x , y]

【用途】 记忆鼠标的当前位置，其中[x , y]是鼠标相对于屏幕左下角的坐标数据。

20. Profile 属性

【描述】 Profile：[on | off]

【用途】 设置是否启用运行记录文件。

【说明】 on —— 启用运行记录文件；

off —— 不启用运行记录文件。

21. ProfileFile 属性

【描述】 ProfileFile：filename

【用途】 定义 MATLAB 的运行记录文件的文件名。

22. ProfileCount 属性

【描述】 ProfileCount：number

【用途】 定义运行记录文件的数量。

23. ProfileInterval 属性

【描述】 ProfileInterval：number

【用途】 定义运行记录文件的记录间隔时间。

24. ScreenDepth 属性

【描述】 ScreenDepth：number

【用途】 设置屏幕的颜色显示深度。例如，若 number = 8，则可显示 256 种颜色($2^8=256$)。

25. ScreenSize 属性

【描述】 ScreenSize：[left , bottom , width , height]

【用途】 描述屏幕的尺寸大小。注意，屏幕尺寸的计算都是以其左下角为参考点的。

26. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

27. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

28. ShowHiddenHandles 属性

【描述】 ShowHiddenHandles：[on | {off}]

【用途】 设置是否显示隐藏的对象。

【说明】 on —— 显示隐藏的对象；

off —— 不显示隐藏的对象。

29. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

30. Units 属性

【描述】 Units：[inches | centimeters | normalized | points | pixels]

【用途】 本对象中各个图形窗口的位置及大小之计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized ——规格化单位，即屏幕的左下角为(0,0)，右上角为(1,1)；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

31. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

32. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

4.4 Figure 对象的属性

1. BackingStore 属性

【描述】 BackingStore：[{on} | off]

【用途】 设置当图形窗口进行切换时，是否重新绘制图形。

【说明】 on —— 不重新绘制图形；

off —— 重新绘制图形。

2. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

3. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

4. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的“子女”图形对象。

5. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

6. CloseRequestFcn 属性

【描述】 CloseRequestFcn：commands

【用途】 关闭窗口时的处理函数。

7. Color 属性

【描述】 Color：[R G B]或 color string

【用途】设置图形窗口的背景颜色。其中 R,G,B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。

8. Colormap 属性

【描述】 Colormap：[matrix $]_{M\times3}$

【用途】 定义颜色分布表。颜色分布表是一个 M×3 维的矩阵（M 的缺省值是 64），其列向量分别表示 R，G，B 三原色的使用浓度。

9. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

10. CurrentAxes 属性

【描述】 CurrentAxes：handle

【用途】 记忆当前坐标轴的句柄。

11. CurrentCharacter 属性

【描述】 CurrentCharacter：character

【用途】 记忆用户当前所按键的键名。

12. CurrentObject 属性

【描述】 CurrentObject：handle

【用途】 记忆当前操作的图形对象的句柄。

13. CurrentPoint 属性

【描述】 CurrentPoint：[x ，y]

【用途】 记忆当按下鼠标或释放鼠标时，鼠标相对于图形窗口左下角的坐标位置。

14. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

15. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

16. IntegerHandle 属性

【描述】 IntegerHandle：[{on} | off]

【用途】 描述是否使用整数句柄值。

【说明】 on —— 使用整数句柄值；

off —— 不使用整数句柄值。

17. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断;

off —— 不可中断。

18. InvertHardcopy 属性

【描述】 InvertHardcopy: [{on} | off]

【用途】 打印图形时,是否进行颜色反转。

【说明】 on —— 进行颜色反转;

off —— 不进行颜色反转。

19. KeyPressFcn 属性

【描述】 KeyPressFcn: commands

【用途】 按键盘键之后的处理函数。

20. MenuBar 属性

【描述】 MenuBar: [none | {figure}]

【用途】 描述是否在图形窗口中设置菜单栏。

【说明】 none —— 不设置菜单栏;

figure —— 设置菜单栏。

21. MinColormap 属性

【描述】 MinColormap: $[\ \text{matrix}\ ]_{M\times 3}$

【用途】 设置绘图时所用的最少色彩种类。其中矩阵的列向量分别表示 R,G,B 三原色的使用浓度,M 的缺省值是 64。

22. Name 属性

【描述】 Name: string

【用途】 记忆图形窗口的名称。这个名称是图形窗口的数字标号之后的文字内容。

23. NextPlot 属性

【描述】 NextPlot: [{add} | replace | replacechildren]

【用途】 描述使用高级绘图命令(如 plot , plot3 , mesh , surf 等命令)绘图时,使用哪一个图形窗口。

【说明】 add —— 在当前图形窗口中绘图;

replace——在当前图形窗口中绘图,但要先删除窗口中的原有的所有图形对象;

replacechildren —— 在当前图形窗口中绘图,但要先删除窗口中的某个指定的“子女”图形对象。

24. NumberTitle 属性

【描述】 NumberTitle: [{on} | off]

【用途】 描述在图形窗口的标题栏上是否显示图形窗口的编号“Figure No. x”,其中 x 代表图形窗口的编号,x=1,2,3,…。

【说明】 on —— 显示图形窗口的编号;

off —— 不显示图形窗口的编号。

25. PaperUnits 属性

【描述】 PaperUnits: [{inches} | centimeters | normalized | points]

【用途】 描述打印纸的尺寸计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位，即打印纸的左下角为(0,0)，右上角为(1,1)；

points —— 点，1 point = 1/72 in。

26. PaperOrientation 属性

【描述】 PaperOrientation：[{portrait} | landscape]

【用途】 描述图形对象在打印纸上的放置方向。

【说明】 portrait —— 纵向；

landscape —— 横向。

27. PaperPosition 属性

【描述】 PaperPosition：[left , bottom , width , height]

【用途】 描述图形对象在打印纸中的打印位置。注意，坐标位置的计算是以打印纸的左下角为参考点的。

28. PaperPositionMode 属性

【描述】 PaperPositionMode：[auto | {manual}]

【用途】 描述图形对象在打印纸中的打印位置之定位方式。

【说明】 auto —— 自动定位；

manual —— 手工定位。

29. PaperSize 属性

【描述】 PaperSize：[width height]

【用途】 定义打印纸的尺寸大小值，包括其宽度和高度。

30. PaperType 属性

【描述】 PaperType：[{usletter} | uslegal | a3 | a4letter | a5 | b4 | tabloid]

【用途】 定义打印纸的尺寸大小种类。

【说明】 usletter —— 8.5 in×11 in；

uslegal —— 8.5 in×14 in；

a3 —— 297 mm×419 mm；

a4letter(a4) —— 210 mm×419 mm；

a5 —— 148 mm×210 mm；

b4 —— 250 mm×353 mm；

tabloid —— 279 mm×431 mm。

31. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

32. Pointer 属性

【描述】 Pointer：[crosshair | fullcrosshair | {arrow} | ibeam | watch | topl | topr | botl | botr | left | top | right | bottom | circle | cross | fleur | custom]

【用途】 定义鼠标形式的种类。

【说明】 crosshair —— 十字线型鼠标；

fullcrosshair —— 满十字线型鼠标；
arrow —— 箭头型鼠标；
ibeam —— I 型鼠标；
watch —— 砂漏型鼠标；
topl —— 左上向的箭头型鼠标；
topr —— 右上向的箭头型鼠标；
botl —— 左下向的箭头型鼠标；
botr —— 右下向的箭头型鼠标；
left —— 向左的箭头型鼠标；
top —— 向上的箭头型鼠标；
right ——向右的箭头型鼠标；
bottom —— 向下的箭头型鼠标；
circle —— 圆圈型鼠标；
cross —— 交叉路口型鼠标；
fleur —— 十字箭头型鼠标；
custom —— 图像型鼠标。

33. Position 属性

【描述】 Position：[left ，bottom ，width ，height]

【用途】 描述本图形窗口在屏幕中的位置。注意，坐标位置的计算是以屏幕的左下角为参考点的。

34. Renderer 属性

【描述】 Renderer：[{painters} | zbuffer]

【用途】 设置图形渲染方法。

35. RendererMode 属性

【描述】 RendererMode：[{auto} | manual]

【用途】 设置图形渲染方式。

【说明】 auto —— 自动进行图形渲染；
manual —— 人工设置图形渲染方法。

36. Resize 属性

【描述】 Resize：[{on} | off]

【用途】 描述是否可用鼠标改变图形窗口的大小。

【说明】 on —— 可用鼠标改变图形窗口的大小；
off —— 不能用鼠标改变图形窗口的大小。

37. ResizeFcn 属性

【描述】 ResizeFcn：commands

【用途】 改变图形窗口大小后的处理函数。

38. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

39. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

40. SelectionType 属性

【描述】 SelectionType：[normal | extent | alt | open]

【用途】 记忆鼠标的哪一个键被按下。

【说明】 normal —— 鼠标左键被按下(或：双击鼠标左键时，第一次按下鼠标左键)；

extent —— 鼠标右键被按下；

alt —— 鼠标中间键被按下；

open —— 双击鼠标左键时，第二次按下鼠标左键。

41. ShareColors 属性

【描述】 ShareColors：[{on} | off]

【用途】 设置图形存储时是否可用相近颜色代替。

【说明】 on —— 可用相近颜色代替；

off —— 不能用相近颜色代替。

42. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

43. Units 属性

【描述】 Units：[inches | centimeters | normalized | points | {pixels}]

【用途】 本图形窗口中各个图形对象的位置及大小计量用的计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位，即屏幕的左下角为(0,0)，右上角为(1,1)；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

44. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

45. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

46. WindowButtonDownFcn 属性

【描述】 WindowButtonDownFcn：commands

【用途】 在图形窗口中按下鼠标键时的处理函数。

47. WindowButtonMotionFcn 属性：commands

【描述】 WindowButtonMotionFcn

【用途】 在图形窗口中移动鼠标时的处理函数。

48. WindowButtonUpFcn 属性：commands

【描述】 WindowButtonUpFcn

【用途】 在图形窗口中释放鼠标键时的处理函数。

49. WindowStyle 属性

【描述】 WindowStyle：[{normal} | modal]

【用途】 描述图形窗口的风格。

【说明】 normal —— 常规风格；

modal —— 有模式风格。

4.5 Axes 对象的属性

1. AmbientLightColor 属性

【描述】 AmbientLightColor：[R G B]或 color string

【用途】 设置背景光颜色。该属性的缺省值是 [1 1 1],即白色。

2. Box 属性

【描述】 Box：[on | {off}]

【用途】 描述是否封闭 Axes 图形对象的坐标区域,即是否产生封闭的坐标系。

【说明】 on —— 封闭坐标区域；

off —— 不封闭坐标区域。

3. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

4. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

5. CameraPosition 属性

【描述】 CameraPosition：[x y z]

【用途】 设置轴的观察点位置。

6. CameraPositionMode 属性

【描述】 CameraPositionMode：[{auto} | manual]

【用途】 设置观察点位置的选定方式。

【说明】 auto —— 自动设置观察点；

manual —— 人工设置观察点。

7. CameraTarget 属性

【描述】 CameraTarget：[x y z]

【用途】 设置轴的观察目标点位置。

8. CameraTargetMode 属性

【描述】 CameraTargetMode：[{auto} | manual]

【用途】 设置观察目标点位置的选定方式。

【说明】 auto —— 自动设置观察目标点；

manual —— 人工设置观察目标点。

9. CameraUpVector 属性

【描述】 CameraUpVector：[x y z]

【用途】 设置观察点的上向轴向量。

10. CameraUpVectorMode 属性

【描述】 CameraUpVectorMode：[{auto} | manual]

【用途】 设置观察点的上向轴向量的选定方式。

【说明】 auto —— 自动设置观察点的上向轴向量；

manual —— 人工设置观察点的上向轴向量。

11. CameraViewAngle 属性

【描述】 CameraViewAngle：number

【用途】 设置观察点的视野范围。其中的 number 为 0～180 的数字。

12. CameraViewAngleMode 属性

【描述】 CameraViewAngleMode：[{auto} | manual]

【用途】 设置观察点视野范围的选定方式。

【说明】 auto —— 自动设置观察点的视野范围；

manual —— 人工设置观察点的视野范围。

13. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的"子女"图形对象。

14. CLim 属性

【描述】 CLim：[cmin cmax]

【用途】 描述颜色分布表的颜色映射方式。

15. CLimMode 属性

【描述】 CLimMode：[{auto} | manual]

【用途】 记忆颜色映射方式的设定方法。

【说明】 auto —— 自动设置；

manual —— 人工设置。

16. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

17. Color 属性

【描述】 Color：[R G B]或 color string

【用途】 设置本 Axes 图形对象的背景颜色。其中 R,G,B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。其缺省的颜色是：[1 1 1]（白色）。

18. ColorOrder 属性

【描述】 ColorOrder：$[\text{ RGB matrix }]_{m\times 3}$

【用途】 以维数为 m×3 的矩阵定义 m 组颜色，当用户用一条命令同时绘制多个图形时，可依次使用不同的颜色。m 的缺省值是 m = 7，缺省的 ColorOrder 属性值是

$$\begin{bmatrix} 0 & 0 & 1.0000 \\ 0 & 0.5 & 0 \\ 1.0000 & 0 & 0 \\ 0 & 0.7500 & 0.7500 \\ 0.7500 & 0 & 0.7500 \\ 0.7500 & 0.7500 & 0 \\ 0.2500 & 0.2500 & 0.2500 \end{bmatrix}$$

19. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

20. CurrentPoint 属性

【描述】 CurrentPoint：$[\text{ x y z }]_{2\times 3}$

【用途】 当用户按下鼠标按键或释放鼠标按键时，该属性记忆鼠标的当前位置。该矩阵的第 1 列表示 x 坐标，第 2 列表示 y 坐标，第 3 列表示 z 坐标。

21. DataAspectRatio 属性

【描述】 DataAspectRatio：[rx ry rz]

【用途】 设置 X,Y,Z 三个坐标轴单位刻度绘制长度之比。

22. DataAspectRatioMode 属性

【描述】 DataAspectRatioMode：[{auto} | manual]

【用途】 设置 X,Y,Z 三个坐标轴单位刻度绘制长度之比的选定方式。

【说明】 auto —— 自动设置单位刻度绘制长度之比；

manual —— 人工设置单位刻度绘制长度之比。

23. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

24. DrawMode 属性

【描述】 DrawMode：[{normal} | fast]

【用途】 设置图形重绘时的绘图顺序。

【说明】 normal —— 以由后向前的次序重绘各个图形对象；

fast —— 以各个图形对象的产生次序重绘。

25. FontAngle 属性

【描述】 FontAngle：[{normal} | italic | oblique]

【用途】 描述坐标轴标记文字的倾斜性。

【说明】 normal —— 不倾斜；

italic —— 使用粗斜体文字；

oblique —— 使用斜体文字。

26. FontName 属性

【描述】 FontName：string

【用途】 描述坐标轴标记文字的字形名称。字形名称有：Helvetica，Times，Courier，Symbol，Arial 等，FontName 属性的缺省字型是：Helvetica。

27. FontSize 属性

【描述】 FontSize：number

【用途】 描述坐标轴标记文字的尺寸大小。其缺省值为 10。

28. FontUnits 属性

【描述】 FontUnits：[inches | centimeters | normalized | {points} | pixels]

【用途】 坐标轴标记文字的尺寸大小之计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位，即坐标轴的左下角为(0,0)，右上角为(1,1)；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

29. FontWeight 属性

【描述】 FontWeight：[light | {normal} | demi | bold]

【用途】 描述坐标轴标记文字的字体粗细。

【说明】 light —— 细体字；

normal —— 正常体；

demi —— 次粗体；

bold —— 粗体。

30. GridLineStyle 属性

【描述】 GridLineStyle：[- | -- | {:} | -. | none]

【用途】 定义网格线的线型。

【说明】 - —— 实线；

-- —— 虚线；

: —— 点线；

-. —— 点划线；

none —— 无网格线。

31. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；
off —— 关闭；
callback —— 使用回调函数。

32. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；
off —— 不可中断。

33. Layer 属性

【描述】 Layer：[top | {bottom}]

【用途】 定义坐标轴是位于顶层还是底层。

【说明】 top —— 位于顶层；
bottom —— 位于底层。

34. LineStyleOrder 属性

【描述】 LineStyleOrder：vector

【用途】 定义多个 Axes 图形对象的线型次序。

35. LineWidth 属性

【描述】 LineWidth：number

【用途】 记忆坐标轴及其刻度线的线条宽度。该属性的缺省值是 0.5。

36. NextPlot 属性

【描述】 NextPlot：[add | {replace} | replacechildren]

【用途】 描述使用高级绘图命令(如 plot，plot3，mesh，surf 等命令)绘图时，使用哪一个坐标轴。

【说明】 add —— 在当前坐标轴中绘图；
replace——在当前坐标轴中绘图，但要先删除坐标轴中的原有的所有图形对象；
replacechildren —— 在当前坐标轴中绘图，但要先删除坐标轴中的某个指定的“子女”图形对象。

37. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

38. PlotBoxAspectRatio 属性

【描述】 PlotBoxAspectRatio：[rx ry rz]

【用途】 设置绘制立方体时各轴的比例。

39. PlotBoxAspectRatioMode 属性

【描述】 PlotBoxAspectRatioMode：[{auto} | manual]

【用途】 设置绘制立方体时各轴的比例选定方式。

【说明】 auto —— 自动设置各轴的比例；

manual —— 人工设置各轴的比例。

40. Projection 属性

【描述】 Projection：[{orthographic} | perspective]

【用途】 设置图形的投影方式。

【说明】 orthographic —— 正交投影；

perspective —— 透视投影。

41. Position 属性

【描述】 Position：[left , bottom , width , height]

【用途】 描述坐标轴在图形窗口中的位置。注意，坐标位置的计算是以当前图形窗口的左下角为参考点的。

42. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

43. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

44. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

45. TickLength 属性

【描述】 TickLength：[length2 length3]

【用途】 定义坐标轴刻度线的长度值。这里的长度值是以坐标轴长度的百分比来表示的。上面的描述中，length2 表示二维刻度线的长度，length3 表示三维刻度线的长度。该属性的缺省值是[0.0100 0.0250]。

46. TickDir 属性

【描述】 TickDir：[{in} | out]

【用途】 描述坐标轴刻度线的方向。

【说明】 in —— 向内；

out —— 向外。

47. TickDirMode 属性

【描述】 TickDirMode：[{auto} | manual]

【用途】 描述坐标轴刻度线之方向的确定方式。

【说明】 auto —— 自动确定坐标轴刻度线之方向；

manual —— 人工确定坐标轴刻度线之方向。

48. Title 属性

【描述】 Title：handle

【用途】 记忆要显示于图形标题处的文字串对象之句柄。

49. Units 属性

【描述】 Units：[inches | centimeters | {normalized} | points | pixels]

【用途】 坐标轴的尺寸及位置计量之计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

50. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

51. View 属性

【描述】 View：[az el]

【用途】 记忆视线的方向。其中参数 az 表示在 XY 平面上，观测点与坐标原点连线的投影与－Y 轴的夹角大小（右手坐标系）；参数 el 表示观测点与坐标原点的连线与 XY 平面的夹角大小（el 取正值时，表示观测点位于 XY 平面之上，el 取负值时，表示观测点位于 XY 平面之下）。

52. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

53. XColor 属性

【描述】 XColor：[R G B] 或 color string

【用途】 设置 X 轴、X 轴上的刻度线及符号、沿 X 轴的各条纵向网格线的颜色。其中 R，G，B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。

54. XDir 属性

【描述】 XDir：[{normal} | reverse]

【用途】 设置 X 轴的坐标方向。

【说明】 normal —— 正常方向，即由左向右；

reverse —— 反方向，即由右向左。

55. XForm 属性

【描述】 XForm：$[\text{ matrix }]_{4\times4}$

【用途】 记忆三维空间到二维图形的变换矩阵。

56. XGrid 属性

【描述】 XGrid：[on | {off}]

【用途】 设定是否在 X 轴的每一个刻度线处画网格线。

【说明】 on —— 在 X 轴的每一个刻度线处画网格线；

off —— 不在 X 轴的每一个刻度线处画网格线。

57. XLabel 属性

【描述】 XLabel：string

【用途】 设置 X 轴的坐标名称。

58. XAxisLocation 属性

【描述】 XAxisLocation：[top | {bottom}]

【用途】 记忆 X 轴的放置位置。

【说明】 top —— X 轴在顶部；

bottom —— X 轴在底部。

59. XLim 属性

【描述】 XLim：[xmin xmax]

【用途】 记忆 X 轴的范围。其中 xmin 为 X 轴的最小值，xmax 为 X 轴的最大值。

60. XLimMode 属性

【描述】 XLimMode：[{auto} | manual]

【用途】 描述 X 轴坐标范围的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

61. XScale 属性

【描述】 XScale：[{linear} | log]

【用途】 描述 X 轴是线性坐标，还是对数坐标。

【说明】 linear —— 线性坐标；

log —— 对数坐标。

62. XTick 属性

【描述】 XTick：vector

【用途】 定义 X 轴上各个刻度的位置。即在向量 vector 的各个元素处画刻度线，若不设置刻度线，则使 vector = [ ] 即可。

63. XTickLabel 属性

【描述】 XTickLabel：matrix

【用途】 定义 X 轴各个刻度记号的字符串内容。

64. XTickLabelMode 属性

【描述】 XTickLabelMode：[{auto} | manual]

【用途】 描述 X 轴的各个刻度记号的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

65. XTickMode 属性

【描述】 XTickMode：[{auto} | manual]

【用途】 描述 X 轴的各个刻度线位置的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

66. YColor 属性

【描述】 YColor：[R G B] 或 color string

【用途】 设置 Y 轴、Y 轴上的刻度线及符号、沿 Y 轴的各条横向网格线的颜色。其中 R，G，B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。

67. YDir 属性

【描述】 YDir：[{normal} | reverse]

【用途】 设置 Y 轴的坐标方向。

【说明】 normal —— 正常方向，即由下向上；

reverse —— 反方向，即由上向下。

68. YGrid 属性

【描述】 YGrid：[on | {off}]

【用途】 设定是否在 Y 轴的每一个刻度线处画网格线。

【说明】 on —— 在 Y 轴的每一个刻度线处画网格线；

off —— 不在 Y 轴的每一个刻度线处画网格线。

69. YLabel 属性

【描述】 YLabel：string

【用途】 设置 Y 轴的坐标名称。

70. YAxisLocation 属性

【描述】 YAxisLocation：[{left} | right]

【用途】 记忆 Y 轴的放置位置。

【说明】 left —— Y 轴在左边；

right —— Y 轴在右边。

71. YLim 属性

【描述】 YLim：[ymin ymax]

【用途】 记忆 Y 轴的范围。其中 ymin 为 Y 轴的最小值，ymax 为 Y 轴的最大值。

72. YLimMode 属性

【描述】 YLimMode：[{auto} | manual]

【用途】 描述 Y 轴坐标范围的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

73. YScale 属性

【描述】 YScale：[{linear} | log]

【用途】 描述 Y 轴是线性坐标，还是对数坐标。

【说明】 linear —— 线性坐标；

log —— 对数坐标。

74. YTick 属性

【描述】 YTick：vector

【用途】 定义 Y 轴上各个刻度的位置。即在向量 vector 的各个元素处画刻度线，若不设置刻度线，则使 vector = [] 即可。

75. YTickLabel 属性

【描述】 YTickLabel：matrix

【用途】 定义 Y 轴各个刻度记号的字符串内容。

76. YTickLabelMode 属性

【描述】 YTickLabelMode：[{auto} | manual]

【用途】 描述 Y 轴的各个刻度记号的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

77. YTickMode 属性

【描述】 YTickMode：[{auto} | manual]

【用途】 描述 Y 轴的各个刻度线位置的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

78. ZColor 属性

【描述】 ZColor：[R G B] 或 color string

【用途】 设置 Z 轴、Z 轴上的刻度线及符号、沿 Z 轴的各条网格线的颜色。其中 R,G,B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。

79. ZDir 属性

【描述】 ZDir：[{normal} | reverse]

【用途】 设置 Z 轴的坐标方向。

【说明】 normal —— 正常方向，即由下向上；

reverse —— 反方向，即由上向下。

80. ZGrid 属性

【描述】 ZGrid：[on | {off}]

【用途】 设定是否在 Z 轴的每一个刻度线处画网格线。

【说明】 on —— 在 Z 轴的每一个刻度线处画网格线；

off —— 不在 Z 轴的每一个刻度线处画网格线。

81. ZLabel 属性

【描述】 ZLabel：string

【用途】 设置 Z 轴的坐标名称。

82. ZLim 属性

【描述】 ZLim：[zmin zmax]

【用途】 记忆 Z 轴的范围。其中 zmin 为 Z 轴的最小值，zmax 为 Z 轴的最大值。

83. ZLimMode 属性

【描述】 ZLimMode：[{auto} | manual]

【用途】 描述 Z 轴坐标范围的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

84. ZScale 属性

【描述】 ZScale：[{linear} | log]

【用途】 描述 Z 轴是线性坐标，还是对数坐标。

【说明】 linear —— 线性坐标；

log —— 对数坐标。

85. ZTick 属性

【描述】 ZTick：vector

【用途】 定义 Z 轴上各个刻度的位置。即在向量 vector 的各个元素处画刻度线，若不设置刻度线，则使 vector = [ ] 即可。

86. ZTickLabel 属性

【描述】 ZTickLabel：matrix

【用途】 定义 Z 轴各个刻度记号的字符串内容。

87. ZTickLabelMode 属性

【描述】 ZTickLabelMode：[{auto} | manual]

【用途】 描述 Z 轴的各个刻度记号的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

88. ZTickMode 属性

【描述】 ZTickMode：[{auto} | manual]

【用途】 描述 Z 轴的各个刻度线位置的设置方式。

【说明】 auto —— 自动设置；

manual —— 人工设置。

4.6 Line 对象的属性

1. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

2. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

3. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的“子女”图形对象。

4. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

5. Color 属性

【描述】 Color：[R G B] 或 color string

【用途】 设置线条的颜色。其中 R,G,B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。

6. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

7. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

8. EraseMode 属性

【描述】 EraseMode：[{normal} | background | xor | none]

【用途】 设置线条的擦除方式，即线条的显示效果。

【说明】 normal —— 正常显示方式；

background —— 以背景颜色显示(常用于擦除原有的对象)；

xor —— 以异或的结果颜色显示；

none —— 不对原有颜色进行擦除。

9. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

10. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

11. LineStyle 属性

【描述】 LineStyle：[{—} | —— | : | —. | none]

【用途】 设置线条的线型。

【说明】 — —— 实线；

—— —— 虚线；

: —— 点线；

一. —— 点划线；

none —— 不画线。

12. LineWidth 属性

【描述】 LineWidth：number

【用途】 设置线条的宽度。该属性的缺省值是 0.5。

13. Marker 属性

【描述】 Marker：[+ | o | * | . | x | square | diamond | v | ˆ | > | < | pentagram | hexagram | {none}]

【用途】 设置画线时对各个数据点的标记记号。

【说明】 + —— 用加号(＋)标记各个数据点；

o —— 用圆圈(o)标记各个数据点；

* —— 用星号(*)标记各个数据点；

· —— 用点号(·)标记各个数据点；

x —— 用叉号(×)标记各个数据点；

square ——用小方块(□)标记各个数据点；

diamond ——用小菱形(◇)标记各个数据点；

v ——用倒三角符号(▽)标记各个数据点；

ˆ ——用正三角符号(△)标记各个数据点；

> ——用向右的三角符号(▷)标记各个数据点；

< ——用向左的三角符号(◁)标记各个数据点；

pentagram ——用五角星符号(☆)标记各个数据点；

hexagram ——用六角星符号(○)标记各个数据点；

none —— 不标记各个数据点。

14. MarkerSize 属性

【描述】 MarkerSize：number

【用途】 设置数据点标记记号的尺寸大小。该属性的缺省值是 6 points。

15. MarkerEdgeColor 属性

【描述】 MarkerEdgeColor：[none | {auto}] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的边沿颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

16. MarkerFaceColor 属性

【描述】 MarkerFaceColor：[{none} | auto] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的表面颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

17. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

18. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

19. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

20. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

21. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

22. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

23. XData 属性

【描述】 XData：vector

【用途】 记忆各个数据点的 x 坐标值。

24. YData 属性

【描述】 YData：vector

【用途】 记忆各个数据点的 y 坐标值。

25. ZData 属性

【描述】 ZData：vector

【用途】 记忆各个数据点的 z 坐标值。要注意的是，对于二维图形，该属性值为一个空矩阵[ ]。

4.7 Patch 对象的属性

1. AmbientStrength 属性

【描述】 AmbientStrength：number

【用途】 设置背景光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.3。

2. BackFaceLighting 属性

【描述】 BackFaceLighting：[unlit | lit | {reverselit}]

【用途】 设置多边形的背光控制方法。

【说明】 unlit —— 无光照；

lit —— 光照；

reverselit —— 反光照。

3. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

4. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

5. CData 属性

【描述】 CData：vector

【用途】 记忆多边形各个顶点的颜色值。要说明的是，只有当 Patch 对象的 EdgeColor 属性或 FaceColor 属性设置为“interp”或“flat”时，CData 属性才有意义。

6. CDataMapping 属性

【描述】 CDataMapping：[direct | {scaled}]

【用途】 定义多边形各个顶点颜色值的映射方法。

【说明】 direct —— 直接映射；

scaled —— 分刻度影射。

7. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的“子女”图形对象。

8. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

9. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

10. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

11. DiffuseStrength 属性

【描述】 DiffuseStrength：number

【用途】 设置漫射光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.6。

12. EdgeColor 属性

【描述】 EdgeColor：[none | flat | interp] 或 [R G B] 或 color string

【用途】 设置多边形边界线条的颜色或颜色确定方式。其中R,G,B为0～1之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0表示不用该原色，而1则表示用最大浓度的该原色。该属性的缺省值是 [0 0 0]，即黑色。

【说明】 none —— 边界线无颜色，即不画出边界线；
flat —— 平面式分布颜色；
interp —— 插补式分布颜色。

13. EdgeLighting 属性

【描述】 EdgeLighting：[{none} | flat | gouraud | phong]

【用途】 设置多边形的边的色彩确定方法。

14. EraseMode 属性

【描述】 EraseMode：[{normal} | background | xor | none]

【用途】 设置 Patch 对象的擦除方式，即 Patch 对象的显示效果。

【说明】 normal —— 正常显示方式；
background —— 以背景颜色显示（常用于擦除原有的对象）；
xor —— 以异或的结果颜色显示；
none —— 不对原有颜色进行擦除。

15. FaceColor 属性

【描述】 FaceColor：[none | flat | interp] 或 [R G B] 或 color string

【用途】 设置多边形封闭区域内的填充方式或填充颜色。其中R,G,B为0～1之间的一个数值，分别表示红色、绿色、兰色蓝原色的浓度，0表示不用该原色，而1则表示用最大浓度的该原色。该属性的缺省值是 [0 0 0]，即黑色。

【说明】 none —— 无填充颜色，即只画出多边形的边界线，而不进行填充；
flat —— 平面式分布颜色；
interp —— 插补式分布颜色。

16. FaceLighting 属性

【描述】 FaceLighting：[none | {flat} | gouraud | phong]

【用途】 设置多边形的表面色彩确定方法。

17. Faces 属性

【描述】 Faces：matrix

【用途】 设置多边形的面数据。

18. FaceVertexCData 属性

【描述】 FaceVertexCData：matrix

【用途】 设置多边形顶点的颜色数据。

19. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；
off —— 关闭；

callback —— 使用回调函数。

20. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

21. LineStyle 属性

【描述】 LineStyle：[{－} | －－ | ： | －. | none]

【用途】 设置多边形边界线条的线型。

【说明】 － —— 实线；

－－ —— 虚线；

： —— 点线；

－. —— 点划线；

none —— 不画线。

22. LineWidth 属性

【描述】 LineWidth：number

【用途】 设置多边形边界线条的宽度。该属性的缺省值是 0.5。

23. Marker 属性

【描述】 Marker：[＋ | o | * | . | x | square | diamond | v | ^ | ＞ | ＜ | pentagram | hexagram | {none}]

【用途】 设置多边形边界线条的各个数据点的标记记号。

【说明】 ＋ —— 用加号(＋)标记各个数据点；

o —— 用圆圈(○)标记各个数据点；

* —— 用星号(*)标记各个数据点；

. —— 用点号(·)标记各个数据点；

x —— 用叉号(×)标记各个数据点；

square ——用小方块(□)标记各个数据点；

diamond ——用小菱形(◇)标记各个数据点；

v ——用倒三角符号(▽)标记各个数据点；

^ ——用正三角符号(△)标记各个数据点；

＞ ——用向右的三角符号(▷)标记各个数据点；

＜ ——用向左的三角符号(◁)标记各个数据点；

pentagram ——用五角星符号(☆)标记各个数据点；

hexagram ——用六角星符号(○)标记各个数据点；

none —— 不标记各个数据点。

24. MarkerEdgeColor 属性

【描述】 MarkerEdgeColor：[none | {auto}] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的边沿颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

25. MarkerFaceColor 属性

【描述】 MarkerFaceColor：[{none} | auto] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的表面颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

26. MarkerSize 属性

【描述】 MarkerSize：number

【用途】 设置数据点标记记号的尺寸大小。该属性的缺省值是 6 points。

27. NormalMode 属性

【描述】 NormalMode：[{auto} | manual]

【用途】 设置顶点法向量的确定方式。

【说明】 auto —— 自动设置顶点法向量；

manual —— 人工设置顶点法向量。

28. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的"双亲"图形对象。

29. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

30. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

31. SpecularStrength 属性

【描述】 SpecularStrength：number

【用途】 设置反射光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.9。

32. SpecularExponent 属性

【描述】 SpecularExponent：number

【用途】 设置反射指数。该属性的缺省值是 10。

33. SpecularColorReflectance 属性

【描述】 SpecularColorReflectance：number

【用途】 设置反射率。该属性的缺省值是 1。

34. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

35. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

36. VertexNormals 属性

【描述】 VertexNormals：matrix

【用途】 设置顶点法向量。

37. Vertices 属性

【描述】 Vertices：matrix

【用途】 记忆多边形的各个顶点的坐标。

38. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

39. XData 属性

【描述】 XData：vector

【用途】 记忆多边形边界线条的各个数据点的 x 坐标值。

40. YData 属性

【描述】 YData：vector

【用途】 记忆多边形边界线条的各个数据点的 y 坐标值。

41. ZData 属性

【描述】 ZData：vector

【用途】 记忆多边形边界线条的各个数据点的 z 坐标值。要注意的是，对于二维图形，该属性值为一个空矩阵[ ]。

4.8 Surface 对象的属性

1. AmbientStrength 属性

【描述】 AmbientStrength：number

【用途】 设置背景光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.3。

2. BackFaceLighting 属性

【描述】 BackFaceLighting：[unlit | lit | {reverselit}]

【用途】 设置多边形的背光控制方法。

【说明】 unlit —— 无光照；

lit —— 光照；

reverselit —— 反光照。

3. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

4. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

5. CData 属性

【描述】 CData：matrix

【用途】 记忆 Surface 图形对象的 Zdata 属性中每个元素所对应的颜色值。要说明的是，只有当 Surface 对象的 EdgeColor 属性或 FaceColor 属性设置为"interp"或"flat"时，CData 属性才有意义。

6. CDataMapping 属性

【描述】 CDataMapping：[direct | {scaled}]

【用途】 定义 Surface 图形对象的颜色值映射方法。

【说明】 direct —— 直接映射；

scaled —— 分刻度影射。

7. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的"子女"图形对象。

8. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

9. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

10. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

11. DiffuseStrength 属性

【描述】 DiffuseStrength：number

【用途】 设置漫射光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.6。

12. EdgeColor 属性

【描述】 EdgeColor：[none | flat | interp] 或 [R G B] 或 color string

【用途】 设置 Surface 图形对象的各个组成四边形边界线条的颜色或颜色确定方式。其中 R,G,B 为 0～1 之间的 1 个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。该属性的缺省值是[0 0 0]，即黑色。

【说明】 none —— 边界线无颜色，即不画出边界线；

flat —— 平面式分布颜色；

interp —— 插补式分布颜色。

13. EdgeLighting 属性

【描述】 EdgeLighting：[{none} | flat | gouraud | phong]

【用途】 设置多边形的边的色彩确定方法。

14. EraseMode 属性

【描述】 EraseMode：[{normal} | background | xor | none]

【用途】 设置 Surface 对象的擦除方式，即 Surface 对象的显示效果。

【说明】 normal —— 正常显示方式；

background —— 以背景颜色显示（常用于擦除原有的对象）；

xor —— 以异或的结果颜色显示；

none —— 不对原有颜色进行擦除。

15. FaceColor 属性

【描述】 FaceColor：[none | flat | interp] 或 [R G B] 或 color string

【用途】 设置 Surface 图形对象的各个组成四边形的填充方式或填充颜色。其中 R、G、B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。该属性的缺省值是 flat。

【说明】 none —— 无填充颜色，即只画出多边形的边界线，而不进行填充；

flat —— 平面式分布颜色；

interp —— 插补式分布颜色。

16. FaceLighting 属性

【描述】 FaceLighting：[none | {flat} | gouraud | phong]

【用途】 设置多边形的表面色彩确定方法。

17. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

18. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

19. LineStyle 属性

【描述】 LineStyle：[{-} | -- | : | -. | none]

【用途】 设置 Surface 对象的各个组成四边形边界线条的线型。

【说明】 - —— 实线；

-- —— 虚线；

: —— 点线；

-. —— 点划线；

none —— 不画线。

20. LineWidth 属性

【描述】 LineWidth：number

【用途】 设置 Surface 对象的各个组成四边形边界线条的宽度。该属性的缺省值是 0.5。

21. Marker 属性

【描述】 Marker：[+ | o | * | . | x | square | diamond | v | ^ | > | < | pentagram | hexagram | {none}]

【用途】 设置 Surface 对象的各个数据点的标记记号。

【说明】 + —— 用加号(+)标记各个数据点；

o —— 用圆圈(o)标记各个数据点；

* —— 用星号(*)标记各个数据点；

. —— 用点号(·)标记各个数据点；

x —— 用叉号(×)标记各个数据点；

square ——用小方块(□)标记各个数据点；

diamond ——用小菱形(◇)标记各个数据点；

v ——用倒三角符号(▽)标记各个数据点；

^ ——用正三角符号(△)标记各个数据点；

> ——用向右的三角符号(▷)标记各个数据点；

< ——用向左的三角符号(◁)标记各个数据点；

pentagram ——用五角星符号(☆)标记各个数据点；

hexagram ——用六角星符号(○)标记各个数据点；

none —— 不标记各个数据点。

22. MarkerEdgeColor 属性

【描述】 MarkerEdgeColor：[none | {auto}] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的边沿颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

23. MarkerFaceColor 属性

【描述】 MarkerFaceColor：[{none} | auto] 或 [R G B] 或 color string

【用途】 设置数据点标记记号的表面颜色或颜色确定方式。

【说明】 none —— 无色；

auto —— 自动设置颜色。

24. MarkerSize 属性

【描述】 MarkerSize：number

【用途】 设置数据点标记记号的尺寸大小。该属性的缺省值是 6 points。

25. MeshStyle 属性

【描述】 MeshStyle：[{both} | row | column]

【用途】 设置网格曲面的类型。

【说明】 both —— 行和列两个方向都有网格线；

row —— 只有行方向的网格线；

column —— 只有列方向的网格线。

26. NormalMode 属性

【描述】 NormalMode：[{auto} | manual]

【用途】 设置顶点法向量的确定方式。

【说明】 auto —— 自动设置顶点法向量；

manual —— 人工设置顶点法向量。

27. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

28. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

29. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

30. SpecularColorReflectance 属性

【描述】 SpecularColorReflectance：number

【用途】 设置反射率。该属性的缺省值是 1。

31. SpecularExponent 属性

【描述】 SpecularExponent：number

【用途】 设置反射指数。该属性的缺省值是 10。

32. SpecularStrength 属性

【描述】 SpecularStrength：number

【用途】 设置反射光强。其中光强为一个 0～1 之间的数值。该属性的缺省值是 0.9。

33. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

34. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

35. VertexNormals 属性

【描述】 VertexNormals：matrix

【用途】 设置顶点法向量。

36. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

37. XData 属性

【描述】 XData：matrix

【用途】 记忆 Surface 对象的各个数据点的 x 坐标值。

38. YData 属性

【描述】 YData：matrix

【用途】 记忆 Surface 对象的各个数据点的 y 坐标值。

39. ZData 属性

【描述】 ZData：matrix

【用途】 记忆 Surface 对象的各个数据点的 z 坐标值。

4.9 Image 对象的属性

1. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

2. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

3. CData 属性

【描述】 CData：vector

【用途】 记忆 Image 图形对象的颜色值。

4. CDataMapping 属性

【描述】 CDataMapping：[direct | {scaled}]

【用途】 定义 Image 图形对象颜色值的映射方法。

【说明】 direct —— 直接映射；

scaled —— 分刻度影射。

5. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的“子女”图形对象。

6. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

7. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

8. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

9. EraseMode 属性

【描述】 EraseMode：[{normal} | background | xor | none]

【用途】 设置 Image 对象的擦除方式，即 Image 对象的显示效果。

【说明】 normal —— 正常显示方式；

background —— 以背景颜色显示(常用于擦除原有的对象)；

xor —— 以异或的结果颜色显示；

none —— 不对原有颜色进行擦除。

10. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

11. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

12. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的"双亲"图形对象。

13. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

14. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

15. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

16. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

17. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

18. XData 属性

【描述】 XData：matrix

【用途】 记忆 Image 对象的各个数据点的 x 坐标值。

19. YData 属性

【描述】 YData：matrix

【用途】 记忆 Image 对象的各个数据点的 y 坐标值。

4.10 Text 对象的属性

1. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

2. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

3. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的"子女"图形对象。

4. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

5. Color 属性

【描述】 Color：[R G B] 或 color string

【用途】 设置 Text 对象(字符串)的显示颜色。其中 R,G,B 为 0～1 之间的一个数值,分别表示红色、绿色、蓝色三原色的浓度,0 表示不用该原色,而 1 则表示用最大浓度的该原色。该属性的缺省值是 [0 0 0],即黑色。

6. CreateFcn 属性

【描述】 CreateFcn: commands

【用途】 产生本图形对象时的处理函数。

7. DeleteFcn 属性

【描述】 DeleteFcn: commands

【用途】 删除本图形对象时的处理函数。

8. EraseMode 属性

【描述】 EraseMode: [{normal} | background | xor | none]

【用途】 设置字符串的擦除方式,即字符串的显示效果。

【说明】 normal —— 正常显示方式;

background —— 以背景颜色显示(常用于擦除原有的字符串);

xor —— 以异或的结果颜色显示;

none —— 不对原有颜色进行擦除。

9. Editing 属性

【描述】 Editing: [on | {off}]

【用途】 设置 Text 对象是否可进行编辑操作。

【说明】 on —— 可进行编辑操作;

off —— 不能进行编辑操作。

10. FontAngle 属性

【描述】 FontAngle: [{normal} | italic | oblique]

【用途】 描述字符串的倾斜性。

【说明】 normal —— 不倾斜;

italic —— 使用粗斜体文字;

oblique —— 使用斜体文字。

11. FontName 属性

【描述】 FontName: string

【用途】 描述字符串的字形名称。字形名称有:Helvetica,Times,Courier,Symbol,Arial 等,FontName 属性的缺省字型是:Helvetica。

12. FontSize 属性

【描述】 FontSize: number

【用途】 描述字符串的尺寸大小。该属性的缺省值为 10。

13. FontUnits 属性

【描述】 FontUnits: [inches | centimeters | normalized | {points} | pixels]

【用途】 字符串的尺寸大小之计量单位。

【说明】 inches —— in;

centimeters —— cm;

normalized —— 规格化单位；

points —— 点，1 point = 1/72 in；

pixels —— 像素；

14. FontWeight 属性

【描述】 FontWeight：[light | {normal} | demi | bold]

【用途】 描述字符串的字体粗细程度。

【说明】 light —— 细体字；

normal —— 正常体；

demi —— 次粗体；

bold —— 粗体。

15. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

16. HorizontalAlignment 属性

【描述】 HorizontalAlignment：[{left} | center | right]

【用途】 描述字符串相对于其指定显示位置的水平对齐方式。

【说明】 left —— 左对齐；

center —— 居中对齐；

right —— 右对齐。

17. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

18. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

19. Position 属性

【描述】 Position：[x , y] 或 [x , y , z]

【用途】 设置字符串的显示位置。注意，这里的坐标值是相对于当前坐标轴的原点而言的。

20. Rotation 属性

【描述】 Rotation：angle

【用途】 设置字符串的显示方向。注意，显示角度的计算是以逆时针方向为正角度。该属性的缺省值是 0°。

21. Selected 属性

【描述】 Selected: [on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择;

off —— 未被选择。

22. SelectionHighlight 属性

【描述】 SelectionHighlight: [{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示;

off —— 不高亮度显示。

23. String 属性

【描述】 String: string

【用途】 记忆所显示的字符串内容。

24. Tag 属性

【描述】 Tag: string

【用途】 设置本图形对象的标记书签(即其名称),该标记书签可用于其句柄的查找。

25. Units 属性

【描述】 Units: [inches | centimeters | normalized | points | pixels | {data}]

【用途】 设置字符串显示位置之计量单位。

【说明】 inches —— in;

centimeters —— cm;

normalized —— 规格化单位;

points —— 点,1 point = 1/72 in;

pixels —— 像素;

data —— 以指定其显示位置的(x,y)坐标值的刻度大小为计量参考单位。

26. UserData 属性

【描述】 UserData: data

【用途】 记忆与本图形对象相关或不相关的用户数据。

27. VerticalAlignment 属性

【描述】 VerticalAlignment: [top | cap | {middle} | baseline | bottom]

【用途】 描述字符串相对于其指定显示位置的垂直对齐方式。

【说明】 top —— 上对齐;

cap —— 以字符顶线对齐;

middle —— 居中对齐;

baseline —— 以字符基线对齐;

bottom —— 下对齐。

28. Visible 属性

【描述】 Visible: [{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见;

off —— 不可见。

4.11 Uimenu 对象的属性

1. Accelerator 属性

【描述】 Accelerator：string

【用途】 描述利用键盘使用菜单的操作方法。

2. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

3. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

4. Callback 属性

【描述】 Callback：string

【用途】 定义当本菜单项被选中时所要执行的命令序列，这个命令序列也叫回调函数。这里的 string 代表一条或多条 MATLAB 命令，也可以是一个变量名称或 M 文件名称。string 的一般书写格式是

```
string = ['command1;'...
          'command2;'...
          ............
          'commandn;']
```

5. Checked 属性

【描述】 Checked：[on | {off}]

【用途】 设置是否在菜单名称前面放置一个 check 标记。

【说明】 on —— 放置 check 标记；

off —— 不放置 check 标记。

6. Children 属性

【描述】 Children：handle

【用途】 描述本图形对象的"子女"图形对象。

7. Clipping 属性

【描述】 Clipping：[{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理；

off —— 不进行剪裁处理。

8. CreateFcn 属性

【描述】 CreateFcn：commands

【用途】 产生本图形对象时的处理函数。

9. DeleteFcn 属性

【描述】 DeleteFcn：commands

【用途】 删除本图形对象时的处理函数。

10. Enable 属性

【描述】 Enable：[{on} | off]

【用途】 设置指定的菜单项是否有效，即能否执行该菜单命令。

【说明】 on —— 该菜单项能够被选中；
off —— 该菜单项不能够被选中。

11. ForegroundColor 属性

【描述】 ForegroundColor：[R G B] 或 color string

【用途】 设置菜单项的前景颜色。其中 R,G,B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。该属性的缺省值是 [0 0 0]，即黑色。

12. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；
off —— 关闭；
callback —— 使用回调函数。

13. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；
off —— 不可中断。

14. Label 属性

【描述】 Label：string

【用途】 设置菜单项的名称。

15. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的“双亲”图形对象。

16. Position 属性

【描述】 Position：number

【用途】 设置菜单项的排列顺序。Position 属性的取值是 1，2，3，…等自然数。菜单项的排列顺序是：先左后右，先上后下。

17. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

18. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

19. Separator 属性

【描述】 Separator：[on | {off}]

【用途】 设置是否在该菜单项之前加一个分隔符。

【说明】 on —— 加一个分隔符；

off —— 不加分隔符。

20. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签（即其名称），该标记书签可用于其句柄的查找。

21. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

22. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

4.12 Uicontrol 对象的属性

1. BackgroundColor 属性

【描述】 BackgroundColor：[R G B] 或 color string

【用途】 设置控件的背景颜色。其中 R，G，B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。该属性的缺省值是 [0.7529 0.7529 0.7529]，即灰色。

2. BusyAction 属性

【描述】 BusyAction：[{queue} | cancel]

【用途】 说明系统忙时的处理方法。

【说明】 queue —— 排队；

cancel —— 终止。

3. ButtonDownFcn 属性

【描述】 ButtonDownFcn：commands

【用途】 按钮按下时的处理函数。

4. Callback 属性

【描述】 Callback: string

【用途】 定义当该控件被选中时所要执行的命令序列,这个命令序列也叫回调函数。这里的 string 代表一条或多条 MATLAB 命令,也可以是一个变量名称或 M 文件名称。string 的一般书写格式是

```
string = ['command1;'...
          'command2;'...
          ............
          'commandn;' ]
```

5. Children 属性

【描述】 Children: handle

【用途】 描述本图形对象的"子女"图形对象。

6. Clipping 属性

【描述】 Clipping: [{on} | off]

【用途】 设置当本图形对象超出 Axes 图形对象所指定的区域时是否进行剪裁处理。

【说明】 on —— 进行剪裁处理;

off —— 不进行剪裁处理。

7. CreateFcn 属性

【描述】 CreateFcn: commands

【用途】 产生本图形对象时的处理函数。

8. DeleteFcn 属性

【描述】 DeleteFcn: commands

【用途】 删除本图形对象时的处理函数。

9. Enable 属性

【描述】 Enable: [{on} | off | inactive]

【用途】 设置本控件是否有效,即能否执行本控件的功能。

【说明】 on —— 本控件有效,并且能执行其预定的功能;

off —— 本控件无效,并且呈淡灰色显示;

inactive —— 本控件有效,但是不能执行其预定的功能。

10. FontAngle 属性

【描述】 FontAngle: [{normal} | italic | oblique]

【用途】 设置控件说明字符串的倾斜性。

【说明】 normal —— 不倾斜;

italic —— 使用粗斜体文字;

oblique —— 使用斜体文字。

11. FontName 属性

【描述】 FontName: string

【用途】 设置控件说明字符串的字形名称。字形名称有:Helvetica,Times,Courier,Symbol,Arial,MS Sans Serif 等,FontName 属性的缺省字型是:MS Sans Serif。

12. FontSize 属性

【描述】 FontSize：number

【用途】 设置控件说明字符串的尺寸大小。该属性的缺省值为 8。

13. FontUnits 属性

【描述】 FontUnits：[inches | centimeters | normalized | {points} | pixels]

【用途】 设置控件说明字符串的尺寸大小之计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

14. FontWeight 属性

【描述】 FontWeight：[light | {normal} | demi | bold]

【用途】 设置控件说明字符串的字体粗细程度。

【说明】 light —— 细体字；

normal —— 正常体；

demi —— 次粗体；

bold —— 粗体。

15. ForegroundColor 属性

【描述】 ForegroundColor：[R G B] 或 color string

【用途】 设置控件的前景颜色。其中 R，G，B 为 0～1 之间的一个数值，分别表示红色、绿色、蓝色三原色的浓度，0 表示不用该原色，而 1 则表示用最大浓度的该原色。该属性的缺省值是 [0 0 0]，即黑色。

16. HandleVisibility 属性

【描述】 HandleVisibility：[{on} | callback | off]

【用途】 说明句柄的可见性。

【说明】 on —— 打开；

off —— 关闭；

callback —— 使用回调函数。

17. HorizontalAlignment 属性

【描述】 HorizontalAlignment：[left | {center} | right]

【用途】 设置控件说明字符串在控件中的放置位置的水平对齐方式。

【说明】 left —— 左对齐；

center —— 居中对齐；

right —— 右对齐。

18. Interruptible 属性

【描述】 Interruptible：[{on} | off]

【用途】 说明该对象的回调函数处理过程是否可中断。

【说明】 on —— 可中断；

off —— 不可中断。

19. Max 属性

【描述】 Max：number

【用途】 描述控件的"Value"属性的最大可能取值。这里要说明的是，不同类型的控件，其"Max"属性具有不同的含义。对于滑块控件，本属性表示其数值描述范围的最大值；而对于单选按钮及检查框，本属性的取值为 1；对于列表框和下拉式菜单，本属性表示其所含可选择项目的个数。

20. Min 属性

【描述】 Min：number

【用途】 描述控件的"Value"属性的最小可能取值。这里要说明的是，不同类型的控件，其"Min"属性具有不同的含义。对于滑块控件，本属性表示其数值描述范围的最小值；而对于单选按钮及检查框，本属性的取值为 0；对于列表框和下拉式菜单，本属性的取值为 1。

21. Parent 属性：handle

【描述】 Parent

【用途】 描述本图形对象的"双亲"图形对象。

22. Position 属性

【描述】 Position：[left ，bottom ，width ，height]

【用途】 描述控件在图形窗口中的位置及大小。注意，坐标位置的计算是以当前图形窗口的左下角为参考点的。

23. Selected 属性

【描述】 Selected：[on | off]

【用途】 描述本图形对象是否已被选择上。

【说明】 on —— 已被选择；

off —— 未被选择。

24. SelectionHighlight 属性

【描述】 SelectionHighlight：[{on} | off]

【用途】 设置本图形对象被选择时是否高亮度显示。

【说明】 on —— 高亮度显示；

off —— 不高亮度显示。

25. SliderStep 属性

【描述】 SliderStep：[number1 number2]

【用途】 设置滑块控件的数值变化步长，其中第一个数字表示用鼠标单击滑块系统两边的箭头后其数值的变化步长，第二个数字表示用鼠标单击滑块的滑动轨道后其数值的变化步长。该属性的缺省值是 [0.0100 0.1000]，即变化步长分别是 1% 和 10%。

26. String 属性

【描述】 String：string

【用途】 设置控件的说明字符串。

27. Style 属性

【描述】 Style：[{pushbutton} | radiobutton | checkbox | edit | text | slider | frame | listbox | popupmenu]

【用途】 设置控件的种类。

【说明】 pushbutton —— 命令按钮；

radiobutton —— 单选按钮；

checkbox —— 检查框；

edit —— 编辑框；

text —— 静态文字；

slider —— 滑块；

frame —— 框架；

listbox —— 列表框；

popupmenu —— 下拉式菜单。

28. Tag 属性

【描述】 Tag：string

【用途】 设置本图形对象的标记书签(即其名称)，该标记书签可用于其句柄的查找。

29. Units 属性

【描述】 Units：[inches | centimeters | normalized | points | {pixels}]

【用途】 设置控件的位置及大小之计量单位。

【说明】 inches —— in；

centimeters —— cm；

normalized —— 规格化单位；

points —— 点，1 point = 1/72 in；

pixels —— 像素。

30. UserData 属性

【描述】 UserData：data

【用途】 记忆与本图形对象相关或不相关的用户数据。

31. Value 属性

【描述】 Value：number

【用途】 记忆控件的当前取值。这里要说明的是，不同类型的控件，其“Value”属性具有不同的含义。对于滑块控件，本属性表示其所在位置的数值描述；而对于单选按钮及检查框，当它们处于被选中的状态时(on 状态)，本属性的取值为 1，当它们处于未被选中的状态时(off 状态)，本属性的取值为 0；对于列表框和下拉式菜单，本属性的取值表示所选项目在全部可选项目中的位置。

32. Visible 属性

【描述】 Visible：[{on} | off]

【用途】 设置本图形对象是否可见。

【说明】 on —— 可见；

off —— 不可见。

第 5 章　MATLAB 环境下的图形用户界面设计技术

自从 Microsoft 公司所开发的 Windows 系统诞生以来，Windows 系统已成为微机操作系统的主流，Windows 系统的发展也推动着微机操作技术的进步。发展的趋势是，很多以前运行在 DOS 下的优秀软件纷纷开发出相应的 Windows 版，以方便用户的操作。MathWorks 公司也于 1993 年正式推出了 MATLAB 4.0 的微机版，以向 Windows 系统靠近。在 Windows 环境下，不仅使用户操作更统一和方便，而且编程人员在编程时也可使用一些通用的形式来构造应用程序，方便了编程工作。如今，MATLAB 下的应用程序也可设计成类似于 Windows 程序的模式，如可以有菜单、对话框、各种控件等。

与其他计算机语言设计的 Windows 应用程序相类似，在 MATLAB 环境下设计 Windows 应用程序也应先生成一个窗口，需要的话也可给这个窗口配上菜单，还可在窗口中生成对话框，布置各种控件等。本章就介绍在 MATLAB 环境下如何构造这种 Windows 的图形用户界面，具体包括下述内容：

- 图形窗口的生成方法
- 菜单的实现方法
- 对话框的构造方法
- 各种控件的设计技术

5.1　图形窗口的生成方法

在 MATLAB 中，无论是生成一个新的图形窗口，还是打开一个已有窗口，都是用 figure 命令完成的。

5.1.1　图形窗口的创建

- 格式：

```
handle = figure('PropertyName1' , PropertyValue1 , 'PropertyName2', PropertyValue2 ,… )
```

- 功能：

以指定的一组属性值，创建一个新的图形窗口，并返回窗口句柄到变量 handle。

- 说明：

① 上述命令格式中，PropertyValue1 是打开窗口时对窗口的 PropertyName1 属性所设定的属性值，PropertyValue2 是打开窗口时对窗口的 PropertyName2 属性所设定的属性值，……

② 由上述命令所生成的窗口是 MATLAB 的一个图形对象，对其各种属性值还可利用 set 命令进行修改，各种具体属性的内容可参见第 4 章。

③ MATLAB 允许用户同时打开多个的图形窗口。

例如，下面的几条命令可以生成一个新的窗口。这个新窗口的生成过程是：先定义一个不可见的窗口 NewWin，然后用 set 命令来设置其各种属性值，最后再用 set 命令将该窗口设置为可见的。

```
》NewWin = figure('visible' , 'off');
》set( NewWin , 'Color' , [1,1,0] , 'Position', [100,100,400,400] , ...
       'Name', 'Application Window', 'NumberTitle' , 'off' , ...
       'Menubar','none');
》set( NewWin , 'Visible', 'on')
```

这里之所以先将窗口设置为不可见的，然后再将其设置为可见的，主要是为了将窗口其他属性的设置过程隐藏起来。其实，上面这几条语句完全可以由如下所示的一条语句来代替，两者的效果完全一致。

```
》NewWin = figure('Color' , [1,1,0] , 'Position', [100,100,400,400] , ...
                  'Name' , 'Application Window', 'NumberTitle' , 'off', ...
                  'Menubar', 'none');
```

又例如，下面的几条命令在屏幕的中间产生一个宽 300 个像素、高 200 个像素的窗口。

```
》ScreenSize = get( 0 , 'ScreenSize' )
ScreenSize =
      1       1     800     600
》NewWin = figure('Name' , 'Application Window', 'NumberTitle', 'off', ...
          'Position',[(ScreenSize(3)-300) * 0.5,(ScreenSize(4)-200) * 0.5,300,
          200] )
NewWin =
      1
```

5.1.2 设置当前窗口

MATLAB 的一些绘图命令(如 plot 命令等)都是在当前窗口中执行的，所以当用户打开了多个图形窗口时，就需要确定哪一个窗口是当前窗口。

- 格式：

  figure (handle)

- 功能：

将句柄为 handle 的图形窗口设置为当前窗口。

· 说明：

若命令所引用的窗口句柄 handle 不存在，则该命令可以为这一句柄生成一个新的窗口，并将所生成的窗口设置为当前窗口。

5.1.3 图形窗口的关闭

对于一些不再使用的图形窗口，用户可以将其关闭，以节约程序运行时所占用的内存。关闭图形窗口的命令是 close。

· 格式：

```
close ( handle )
```

· 功能：

关闭句柄 handle 所指定的图形窗口。

5.2 菜单的实现技术

菜单(menu)是用户图形界面的一个重要组成部分。菜单一般位于图形窗口的最上面(窗口标题之下)，并往往是构成一种下拉式的菜单，用户通过鼠标或键盘可以选择让程序去执行某个菜单项所指定的功能。菜单也是 MATLAB 所定义的一个图形对象，也有很多属性。MATLAB 既提供了生成标准的下拉式菜单的命令 Uimenu，又提供了生成简易菜单的命令 menu，choices。

5.2.1 下拉式菜单的生成

· 格式：

```
MenuHandle = uimenu ( handle,'PropertyName1', 'PropertyValue1' , ...
                      'PropertyName2', 'PropertyValue2', …… )
ItemHandle = uimenu (MenuHandle ,'PropertyName1', 'PropertyValue1', ...
                      'PropertyName2', 'PropertyValue2', …… )
```

· 功能：

在句柄为 handle 的图形窗口中生成一个下拉式的菜单。

· 说明：

① MenuHandle 为在图形窗口中所生成的下拉式菜单的句柄。

② 每个菜单项目都应使用上面第二个格式产生一个句柄值。

③ 在生成这个菜单时，其属性 PropertyName1，PropertyName2…分别被设置为 PropertyValue1，PropertyValue2…等属性值。菜单的显示特性就是由这些属性值所确定的。

(4) 菜单是 MATLAB 定义的一个图形对象，它有很多属性可控制其显示特性。下面为生成菜单时常用到的一些属性(菜单的完整的属性可见第 4 章)：

· Label —— 菜单项目的标记名称。它可以是任意一个字符串。另外，在这个字符串中还可以使用符号“&”，它表示该符号后面的字符在显示时有一个下划线，代表用键盘激活该菜单项的字母键。

• Callback —— 菜单命令的回调函数。它可以是一个函数名，也可以是一组 MATLAB 命令，表示选中该菜单项后系统的响应方法。

• Accelerator —— 菜单的快速执行键。它定义了该菜单命令的快速响应热键，其书写形式为“Alt+F3”，“Ctrl+F5”等。

• Position —— 菜单的位置。它设置了各个菜单项的相对排列位置。

• Separator —— 设置各组菜单项目之间的分隔符。它的默认状态是“off”，当它取值为“on”时，则表示在该菜单项目的上面加一条横线，“—”号作为各组菜单之间的分隔符号。

• Enable —— 设置菜单项的作用状态。它的取值有：on，off。“on”是其默认状态，表示该菜单项处于能够起作用的状态，而“off”则表示该菜单项处于不能够起作用的状态，即无效状态。

例如，下面的示例程序 EX501 可产生图 5-1 所示的菜单。

```
%**************************************
%     程序：EX501
%     功能：菜单程序示例例
%**************************************
ScreenSize = get( 0 , 'ScreenSize');
NewWin = figure( 'Name' , 'Application Window' , ...
  'Position',[(ScreenSize(3)-300) * 0.5,(ScreenSize(4)-200) * 0.5,300,200],...
  'NumberTitle', 'off' , ...
  'MenuBar', 'none' );
FileMenu = uimenu( NewWin, 'Label', '&File' );
EditMenu = uimenu( NewWin, 'Label', '&Edit' );
HelpMenu = uimenu( NewWin, 'Label', '&Help');
ItemNew = uimenu( FileMenu, 'Label', '&New' );
ItemOpen = uimenu( FileMenu, 'Label', '&Open' );
ItemSave = uimenu( FileMenu, 'Label', '&Save' );
ItemSaveAs = uimenu( FileMenu, 'Label', 'Save &As' );
ItemExit = uimenu( FileMenu, 'Label', '&Exit', 'Separator' , 'on' );
ItemFind = uimenu( EditMenu, 'Label', '&Find');
ItemReplace= uimenu( EditMenu, 'Label', '&Replace' );
ItemIndex = uimenu( HelpMenu, 'Label', '&Index' );
ItemContent= uimenu( HelpMenu, 'Label', '&Content' );
```

又例如，下面的示例程序 EX502 中的最后 5 条语句说明了动态扩展或减少菜单项目的方法，并说明了动态决定菜单项目是否起作用的方法，其产生的菜单如图 5-2 所示。

```
%**************************************
%     程序：EX502
%     功能：菜单项目的动态更改程序示例
%**************************************
ScreenSize = get( 0 , 'ScreenSize');
NewWin = figure( 'Name' , 'Application Window' , ...
```

```
    'Position',[(ScreenSize(3)-300) * 0.5,(ScreenSize(4)-200) * 0.5,300,200],...
    'NumberTitle' , 'off' , ...
    'MenuBar', 'none' );
FileMenu = uimenu( NewWin, 'Label', '&File' );
EditMenu = uimenu( NewWin, 'Label', '&Edit' );
HelpMenu = uimenu( NewWin, 'Label', '&Help' );
ItemNew = uimenu( FileMenu, 'Label', '&New' );
ItemOpen = uimenu( FileMenu, 'Label', '&Open');
ItemSave = uimenu( FileMenu, 'Label', '&Save' );
ItemSaveAs = uimenu( FileMenu, 'Label', 'Save &As' );
ItemExit = uimenu( FileMenu, 'Label', '&Exit' , 'Separator' , 'on' );
ItemFind = uimenu( EditMenu, 'Label', '&Find' );
ItemReplace= uimenu( EditMenu, 'Label', '&Replace' );
ItemIndex = uimenu( HelpMenu, 'Label', '&Index' );
ItemContent= uimenu( HelpMenu, 'Label', '&Content' );
set(ItemFind, 'Enable' , 'off');
set(ItemReplace, 'Enable' ,'off');
ItemUndo = uimenu( EditMenu, 'Label', '&Undo', 'separator','on' );
ItemRedo = uimenu( EditMenu, 'Label', '&Redo' );
set(ItemUndo, 'Enable' ,'off');
```

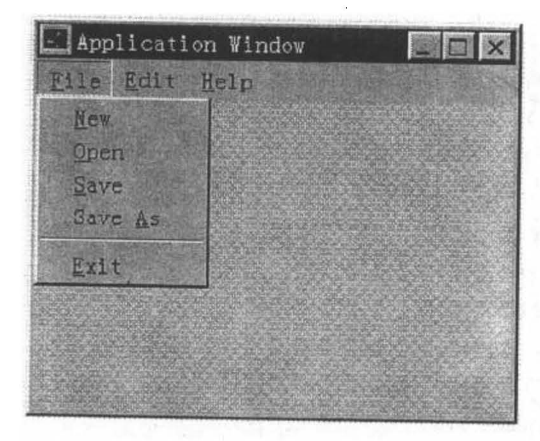


图 5-1 菜单示例

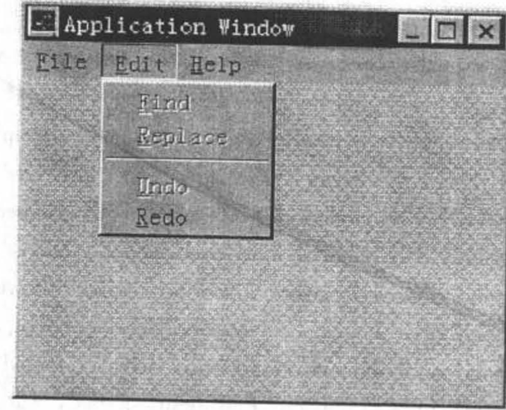


图 5-2 菜单示例

5.2.2 子菜单的生成

如图 5-3 所示，在菜单项中还含有一个菜单，这个菜单称为子菜单。

在 MATLAB 中，子菜单也是由 uimenu 命令来生成的，只是此时子菜单的"双亲"的句柄应为某一菜单项的句柄。生成子菜单的一般语句结构是

```
MenuHandle = uimenu ( handle , 'PropertyName1', 'PropertyValue1' , ...
                              'PropertyName2' , 'PropertyValue2' , ...... )
ItemHandle = uimenu ( MenuHandle , 'PropertyName1', 'PropertyValue1', ...
```

```
                                 'PropertyName2' , 'PropertyValue2', ...... )
    ..................
SubItemHandle = uimenu( ItemHandle , 'PropertyName1' , 'PropertyValue1', ...
                                 'PropertyName2', 'PropertyValue2', ...... )
    ..................
```

例如，下面的示例程序 EX503 可产生出图 5-3 所示的菜单，该菜单就含有子菜单。这个程序的结构是生成一般通用菜单的基本结构形式。

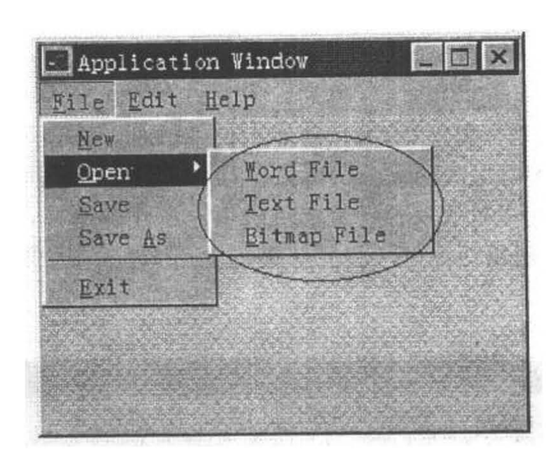


图 5-3　子菜单示例

```
%****************************************
%    程序：EX503
%    功能：通用菜单程序示例
%****************************************
ScreenSize = get( 0 , 'ScreenSize' );
NewWin = figure( 'Name' , 'Application Window' , ...
  'Position',[(ScreenSize(3)-300) * 0.5,(ScreenSize(4)-200) * 0.5,300,200],...
  'NumberTitle' , 'off' , ...
  'MenuBar', 'none' );
FileMenu = uimenu( NewWin, 'Label', '&File' );
EditMenu = uimenu( NewWin, 'Label', '&Edit' );
HelpMenu = uimenu( NewWin, 'Label', '&Help' );
ItemNew = uimenu( FileMenu, 'Label', '&New');
ItemOpen = uimenu( FileMenu, 'Label', '&Open' );
ItemSave = uimenu( FileMenu, 'Label', '&Save' );
ItemSaveAs = uimenu( FileMenu, 'Label', 'Save &As' );
ItemExit = uimenu( FileMenu, 'Label', '&Exit' , 'Separator' , 'on' );
ItemUndo = uimenu( EditMenu, 'Label', '&Undo' );
ItemRedo = uimenu( EditMenu, 'Label', '&ReUndo' );
ItemFind = uimenu( EditMenu, 'Label', '&Find' , 'Separator' , 'on' );
ItemReplace= uimenu( EditMenu, 'Label', '&Replace' );
```

```
ItemIndex = uimenu( HelpMenu, 'Label', '&Index' );
ItemContent= uimenu( HelpMenu, 'Label', '&Content' );
ItemWord = uimenu( ItemOpen, 'Label', '&Word File' );
ItemText = uimenu( ItemOpen, 'Label', '&Text File' );
ItemBmp = uimenu( ItemOpen, 'Label', '&Bitmap File' );
set(ItemUndo, 'Enable' ,'off');
set(ItemRedo, 'Enable' ,'off');
```

5.3 对话框及其实现技术

从本质上说，对话框也是一类窗口；而从功能上说，对话框是用来要求用户输入某些信息或给用户提供某些信息而暂时出现的一个窗口。即对话框是用户和计算机之间进行交互操作的一种手段，通过对话框，用户可以通知计算机一些用户所作的选择，也可以输入一些参数给计算机，并且计算机功能也给用户提供一些信息或各种运行结果等。例如，图 5－4 为一个典型的对话框。

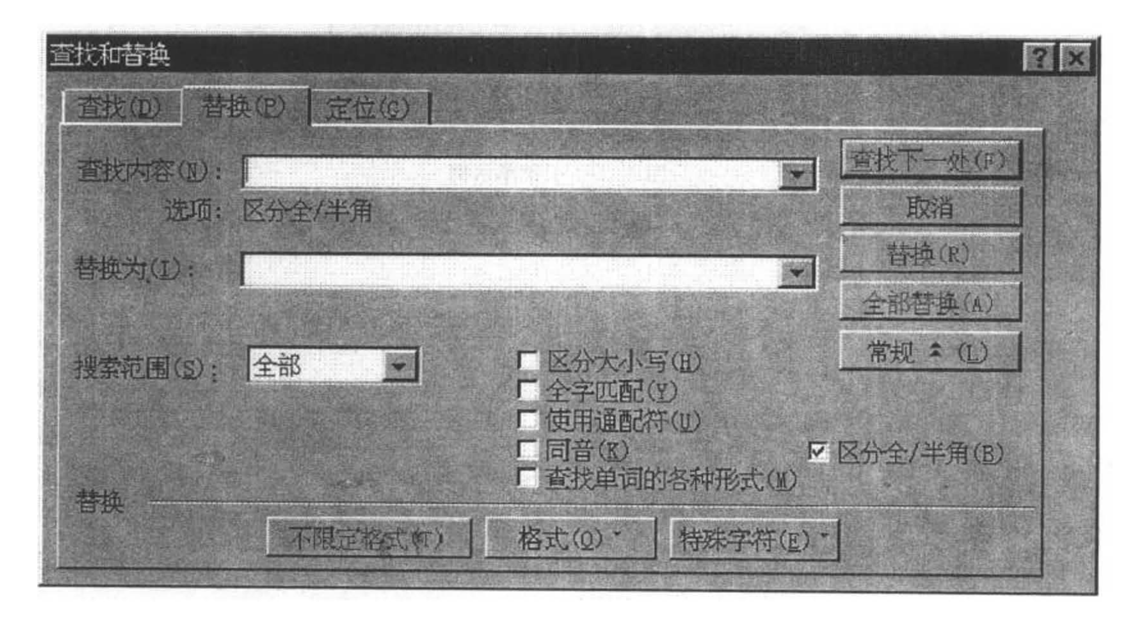


图 5－4 对话框示例

另一方面，对话框的设计又比较复杂。为了给程序员的设计工作提供方便，同时也为了操作界面的规范和统一，MALTAB 系统提供了几个标准对话框的直接调用函数，以简单地实现与 Windows 系统相类似的标准对话框的操作与处理。这些典型的对话框主要有：文件名处理对话框、字体设置对话框和颜色设置对话框。

再者，MATLAB 也提供了一些简单的信息提示性的对话框，如“出错对话框”、“信息对话框”、“警告对话框”等。

5.3.1 普通对话框的实现方法

对话框也是一种窗口，它是用 dialog 命令来生成的。

- 格式：

handle = dialog('PropertyName1', PropertyValue1 , 'PropertyName2' , PropertyValue2 , …)

- 功能：

以指定的一组属性值，创建一个新的对话框，并返回该对话框的句柄到变量 handle。

- 说明：

① 上述命令格式中，PropertyValue1 是创建对话框时对窗口的 PropertyName1 属性所设定的属性值，PropertyValue2 是创建对话框时对窗口的 PropertyName2 属性所设定的属性值……

② 由上述命令所生成的对话框也具有窗口对象的各种属性，并且它使用了下述属性的推荐值：

'BackingStore' —— 'off'
'ButtonDownFun' —— 'if isempty(allchildren(gcbf)) , close(gcbf) , end'
'Colormap' —— [ ]
'Color' —— DefaultUicontrolBackgroundColor
'HandleVisibility' —— 'callback'
'IntegerHandle' —— 'off'
'InvertHardcopy' —— 'off'
'MenuBar' —— MAC = 'figure' others = 'none'
'NumberTitle' —— 'off'
'PagerPositionMode' —— 'auto'
'Resize' —— 'off'
'Visible' —— 'on'
'WindowStyle' —— 'modal'

例如，下面的几条命令在屏幕的中间产生一个宽 400 个像素、高 300 个像素的对话框。这是一个空白的对话框，我们可以给其添加一些控件（具体内容请参见下一节）。

```
》ScreenSize = get( 0 , 'ScreenSize' )
ScreenSize =
      1      1     800     600
》NewDlg = dialog( 'Name' , 'A New Dialogue Window' , . . .
             'Position', [(ScreenSize(3)-400) * 0.5 ,( ScreenSize(4)-300) * 0.5 ,
             400 , 300] )
NewDlg =
      1.0004
```

5.3.2 文件名处理对话框

如图 5-5 所示为 Windows 环境下的一个标准的文件名处理对话框。

很多 Windows 应用程序在牵扯到与文件名相关的处理时，在操作中几乎都使用了图5-5所示的这种文件名处理对话框。

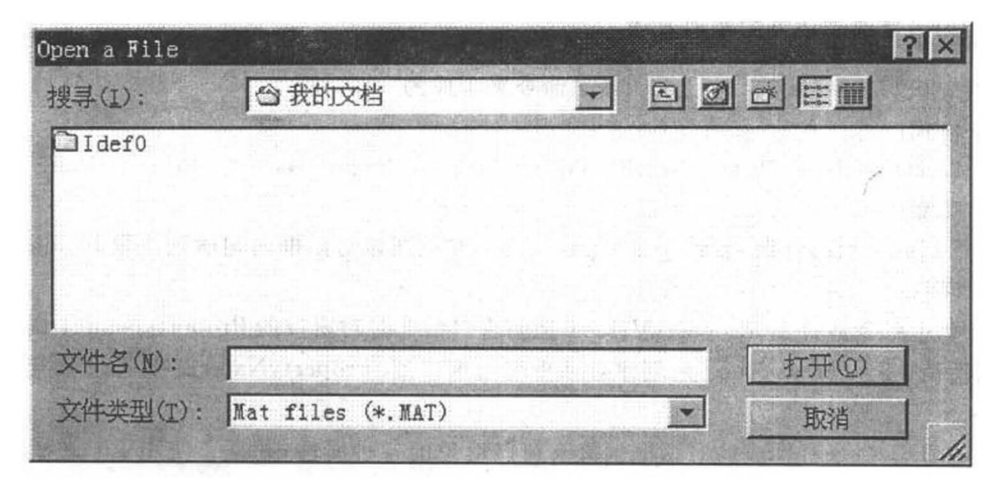


图 5-5 文件名处理对话框

在 MATLAB 环境下，我们可使用 uigetfile()函数来实现这种标准的文件名处理对话框。

· 格式：

[filename , filepath] = uigetfile(FileType , DlgTitle , X , Y)

· 功能：

实现 Windows 环境下的一个标准的文件名处理对话框(打开文件对话框)。

· 说明：

① 参数 FileType 指明了所要处理的文件名的类型；DlgTitle 参数指明了对话框的标题；参数 X,Y 给出了以像素点为单位的对话框在屏幕上的初始位置。

② 当在该对话框进行了一些文件名的定位操作，并按下"打开"(O)按钮之后，系统所查找到的文件之文件名和路径名将分别返回到两个字符串变量 filename 和 filepath 之中。

③ 如果用户按下"取消"按钮，则将取消这个处理文件名的操作并将 0 返回到两个变量 filename 和 filepath 之中。

例如，图 5-5 所示的对话框就是由下述命令产生的。

```
》[ fname , fpath ] = uigetfile( ' *.mat' , 'Open a File' , 100 , 200 )
```

当系统查找到文件名之后，就可对这个文件进行一些相关的处理。例如，下面这段程序的功能就是以用户自己选择的方式打开某个指定的文件，进行一些处理之后，再关闭这个文件。

```
》[ fname , fpath ] = uigetfile( ' *.txt' , 'Open a File' , 100 , 200 )
fname =
test.txt
fpath =
D:\MATLAB\bin\
》fopen( fname , 'r' )
          ............
     》fclose( fname )
```

另外，与 uigetfile()函数相对应，MATLAB 还提供了另一个函数 uiputfile()，该函数可用于打开一个文件(调用标准的文件保存对话框)，以便向该文件输出一些信息。

· 格式：

[filename , filepath] = uiputfile(IniFile , DlgTitle , X , Y)

· 功能：

实现 Windows 环境下的一个标准的保存文件名处理对话框。

· 说明：

① 参数 IniFile 指明了一个初始的文件名或文件名的初始类型；DlgTitle 参数指明了对话框的标题；参数 X,Y 给出了以像素点为单位的对话框在屏幕上的初始位置。

② 当在该对话框进行了一些文件名的定位操作，并按下“保存”(S)按钮之后，系统所指定的文件之文件名和路径名将分别返回到两个字符串变量 filename 和 filepath 之中。

③ 如果用户按下“取消”按钮，则将取消这个处理文件名的操作，并将 0 返回到两个变量 filename 和 filepath 之中。

5.3.3 字体设置对话框

Windows 系统为用户提供了很多种字体，如“Times New Roman”，“Arial”，“宋体”，“黑体”，“楷体”等。使用不同的字体，可美化程序显示结果的外观。进行字体设置时，很多 Windows 应用程序也使用了标准的字体设置对话框，如图 5 - 6 所示

在 MATLAB 环境下，实现这种标准的字体选择对话框的命令是 uisetfont 函数。

· 格式：

Font = uisetfont(Handle , DlgTitle)

· 功能：

弹出一个标准的字体设置对话框，以便用户能够给句柄为 Handle 的字体选择一个新的字体。

· 说明：

① 参数 DlgTitle 指明了对话框的标题。

② 该函数可实现一个标准的字体选择对话框，在该对话框中，用户通过一些操作可指定一个新的字体。

③ 若不提供旧的字体句柄，则该函数可用于系统整体的字体设置。

④ 对于新的字体 Font，我们还可通过 get(Font , 'PropertyName')函数来读取其属性。

⑤ 当在所产生的对话框中进行了一些字体选择操作，并按下"确定"按钮之后，系统将返回所选择字体的句柄值到变量 Font 中；若按下“取消”按钮，则将取消这个字体选择操作，并将 0 返回到变量 Font 中。

例如，图 5 - 6 所示的对话框就是由下述命令实现的。

```
»NewFont = uisetfont( 'Font Select Dialog' )
NewFont =
        11.0001
```

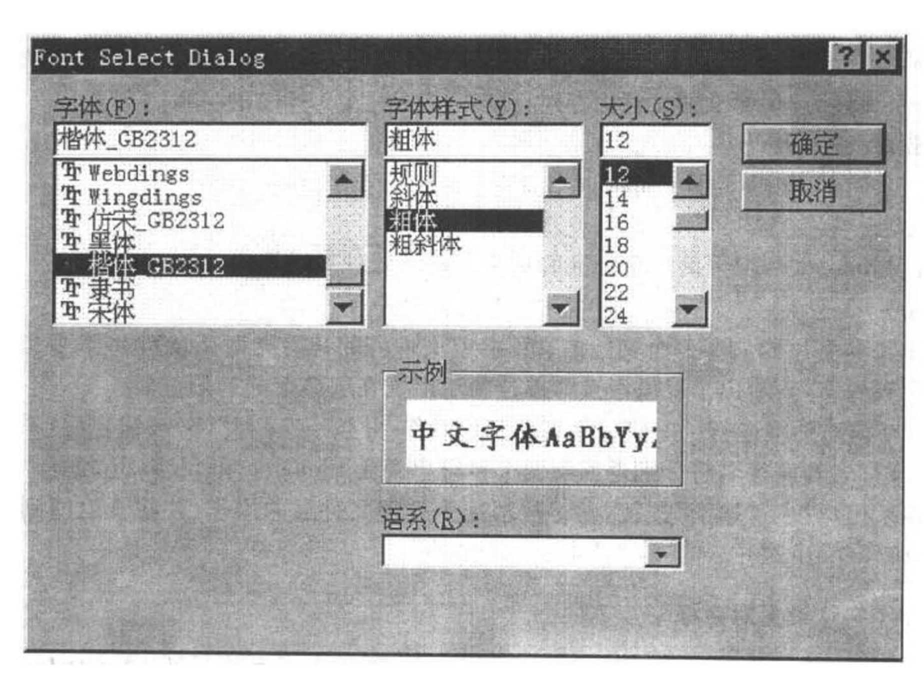


图 5-6　字体设置对话框

5.3.4　颜色设置对话框

在 Windows 应用程序中，当需要进行颜色设置时，往往使用的是图 5-7 所示的标准的颜色设置对话框。该对话框提供了 Windows 系统所预定义的 48 种颜色供用户选用。另外，如果用户还想进一步选用其他颜色，可按“Define Custom Colors”按钮，系统将弹出图 5-8 所示的用户颜色定义对话框，以便使用户通过 Windows 所提供的标准操作方式来定义新的颜色。

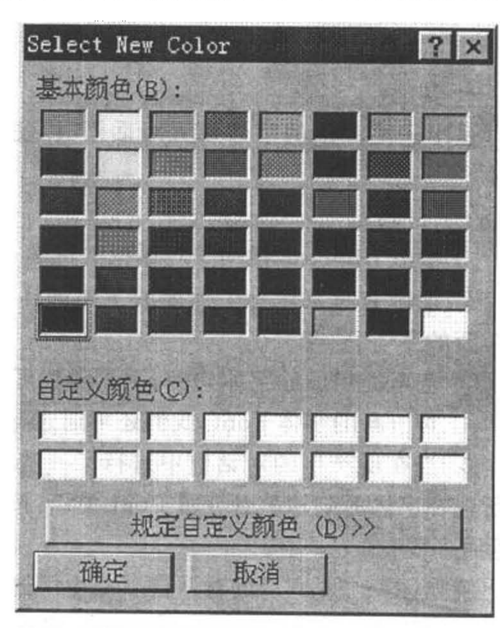


图 5-7　基本颜色设置对话框

MATLAB 也提供了对应于上述操作的函数 uisetcolor()，通过该函数可以使用与上述操作过程类似的方法来设置颜色。

- 格式：

  ColorRGB = uisetcolor(Handle , DlgTitle)

  ColorRGB = uisetcolor(RGB , DlgTitle)

- 功能：

弹出一个标准的颜色设置对话框，以便用户能够为指定的对象或指定的颜色选择一个新的颜色。



- 说明：

① 参数 Handle 指定了一个要设置颜色的图形对象；参数 RGB 是一个三原色数组，它指

定了一个颜色；参数 DlgTitle 指明了对话框的标题。

② 缺省 Handle 及 RGB 时，则该命令用于进行系统整体颜色的设置。

③ 所返回的颜色值 ColorRGB 是一个拥有 3 个元素的向量，它们分别表示新设定的颜色中红色、绿色、蓝色所占的份量，即 ColorRGB = [r g b]，其中各个元素的取值在 0～1 之间。若按下"取消"按钮，则将取消这个颜色设置操作，并将 0 返回到变量 ColorRGB 中。

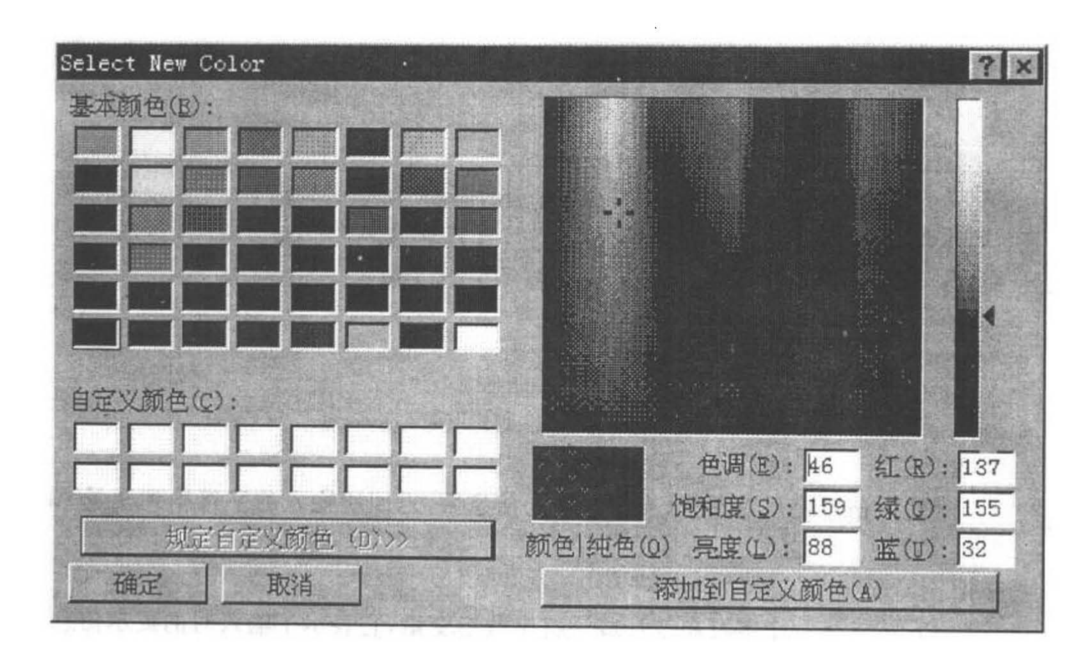


图 5-8 用户扩展颜色设置对话框

例如，图 5-7 和图 5-8 所示对话框就是由下述命令来实现的。

```
»color = uisetcolor( 'Select New Color' )
color =
     1.0000     0.5020     0
```

5.3.5 几种常用的信息提示对话框

MATLAB 给我们提供了几种常用的信息提示用对话框，这些对话框可以方便我们编制 MATLAB 程序。下面我们对这些信息提示用对话框作简要介绍。

一、ErrorDlg（错误信息对话框）

· 格式：

Handle = ErrorDlg(ErrorString)

Handle = ErrorDlg(ErrorString , DlgTitle)

Handle = ErrorDlg(ErrorString , DlgTitle , Replace)

· 功能：

产生一个名称为 DlgTitle 的错误信息对话框，并返回其句柄。

• 说明：

① 参数 ErrorString 是一个字符串变量，它表示了要提示的错误信息。

② 系统弹出该对话框之后，只有按“OK”按钮之后才能取消该对话框。

③ 参数 Replace 的缺省值为'non-modal'。若 Replace = 'replace'，则新产生的错误信息提示对话框可以替换掉原来的同名对话框。参数 Replace 的另一个可能的取值为 'modal'。

例如，下述语句可产生图 5－9 所示的错误信息对话框。

```
》Handle = ErrorDlg( 'An Error Has Been Detected.' , 'Error Information' )
Handle = ·
    11.0015
```

二、InputDlg（输入对话框）

• **格式：**

Answer = InputDlg(Prompt)

Answer = InputDlg(Prompt , DlgTitle)

Answer = InputDlg(Prompt , DlgTitle , LineNo)

Answer = InputDlg(Prompt , DlgTitle , LineNo , DefAnswer)

• **功能：**

产生一个输入信息提示对话框，当用户输入信息后，返回所输入的字符串到变量 Answer 中。

• **说明：**

① 参数 Prompt 是一个字符串变量或字符串数组变量，它表示了输入时的提示信息。

② 参数 DlgTitle 是一个字符串变量，它表示对话框的名称。

③ 参数 LineNo 是一个数值或数组，它说明了各个输入框的行数。

④ 参数 DefAnswer 是一个字符串变量或字符串数组变量，它表示各个输入项目的缺省值。

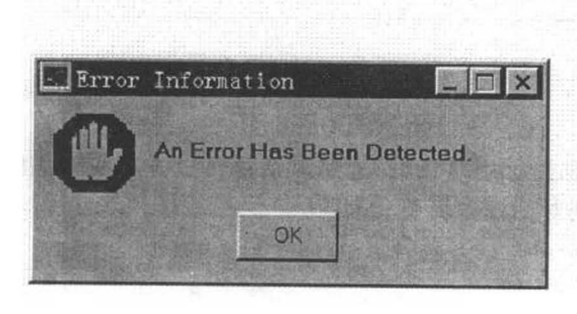


图 5－9　错误信息对话框

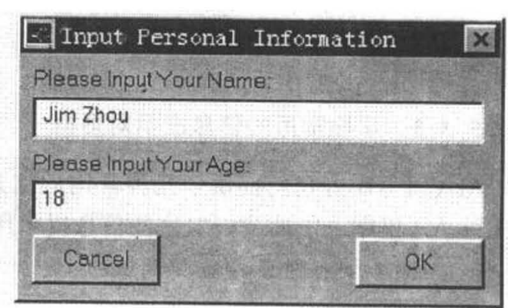


图 5－10　输入对话框

例如，下述语句可产生图 5－10 所示的输入对话框。

```
》PromptStr = { 'Please Input Your Name:' , 'Please Input Your Age:' };
》DefPerson = { 'Jim Zhou', '18'};
》PersonData = InputDlg( PromptStr , 'Input Personal Information' , 1 , DefPerson )
    (按“OK”按钮)
PersonData =
    'Jim Zhou'
    '18'
```

三、MsgBox(消息显示对话框)

· 格式:

MsgBox(Message)

MsgBox(Message , DlgTiTle)

MsgBox(Message , DlgTiTle , Icon)

MsgBox(Message , DlgTiTle , Icon , Mode)

· 功能:

产生一个消息显示对话框。

· 说明:

① 参数 Message 是一个字符串变量或字符串数组变量,它表示了在对话框中要显示的信息内容。

② 参数 DlgTitle 是一个字符串变量,它表示对话框的名称。

③ 参数 Icon 是一个字符串变量,它说明了在对话框中要加入的 Icon(图标)种类。参数Icon的可能取值有:'none'(无图标),'error'(出错图标),'help'(帮助图标),'warn'(警告图标),'custom'(用户自定义图标),其缺省值是'none'。

④ 参数 Mode 是一个字符串变量,它说明所产生的对话框是“modal 型”的,还是“non-modal 型”的,以及是否替换同名对话框。该参数的可能取值有:'modal','non-modal','replace',其缺省值是'non-modal'。

例如,下述 4 条 MsgBox 语句可分别产生图 5-11、图 5-12、图 5-13 和图 5-14 所示的 4 种消息显示对话框。

```
》MsgStr = { 'No This File!' , 'Please Select Another File.' };
》MsgBox( MsgStr )                                                    (图 5-11)
》MsgBox( MsgStr , 'Message Box (warn)' , 'warn' , 'modal' )          (图 5-12)
》MsgBox( MsgStr , 'Message Box (error)' , 'error' , 'modal' )        (图 5-13)
》MsgBox( MsgStr , 'Message Box (help)', 'help' , 'modal' )           (图 5-14)
```

四、QuestDlg(提问对话框)

· 格式:

ButtonName = QuestDlg(QuestionString)

ButtonName = QuestDlg(QuestionString , DlgName)

```
ButtonName = QuestDlg( QuestionString , DlgName , DefButton )
ButtonName = QuestDlg( QuestionString , DlgName , ButtonArray , DefButton )
```

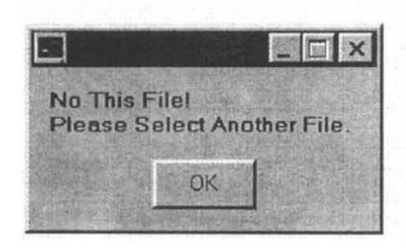


图 5-11　消息显示对话框之一

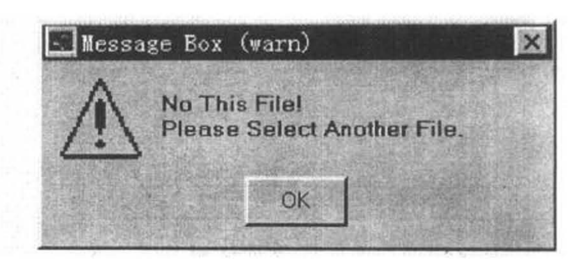


图 5-12　消息显示对话框之二

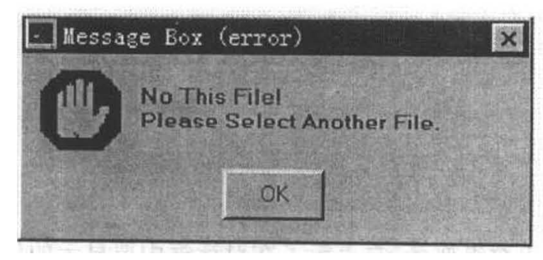


图 5-13　消息显示对话框之三

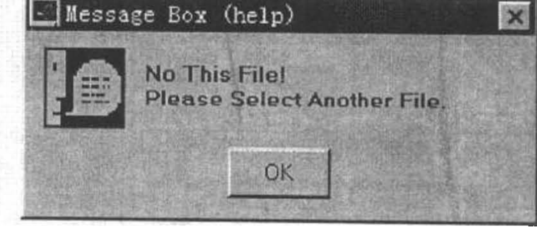


图 5-14　消息显示对话框之四

· 功能：

产生一个用于提问的对话框，该对话框中含有回答该问题的几个答案按钮，用户回答后，可返回用户所选择的回答信息(即返回用户所点击的按钮名称)。

· 说明：

① 该命令的返回变量 ButtonName 是一个字符串变量，其返回内容为用户所点击的提问对话框中的按钮名称。

② 参数 QuestionString 是一个字符串变量，它表示了在对话框中要显示的让用户回答的提问内容。

③ 参数 DlgTitle 是一个字符串变量，它表示对话框的名称。

④ 在缺省情况下，该对话框含有 3 个回答按钮："Yes"，"No"，"Cancel"，其缺省答案按钮是"Yes"。但也可由参数 DefButton 来指定缺省答案按钮。

⑤ 参数 ButtonArray 是一个最多只含有 3 个元素的字符串数组变量，它定义了各个答案按钮的名称。

例如，下述 QuestDlg 语句可产生出图 5-15 所示的提问对话框。

```
» QuestStr = 'Change The Name of This File Now ?';
» ButtonName = QuestDlg( QuestStr , 'Quest Dialog')
   (按"Yes"按钮)
ButtonName =
Yes
```

又例如，下述 QuestDlg 语句可产生出图 5－16 所示的提问对话框。

```
》QuestStr = 'Change The Name of This File Now ?';
》ButtonName = QuestDlg( QuestStr , 'Quest Dialog', ...
                        'Save' , 'Save As', 'Cancel' , 'Cancel' )
   (按"Save As"按钮)
ButtonName =
Save As
```

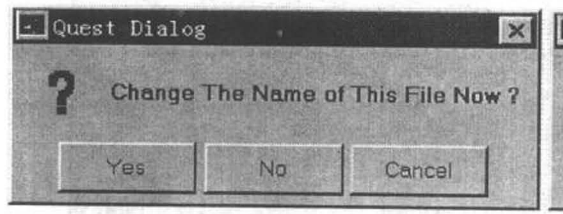


图 5－15　提问对话框

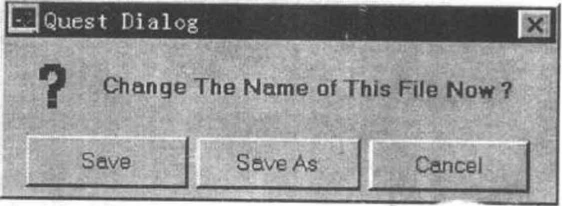


图 5－16　提问对话框

五、WarnDlg(警告信息对话框)

· **格式：**

Handle = WarnDlg(WarnString , DlgTitle)

· **功能：**

产生一个显示警告信息的对话框，并返回其句柄。

· **说明：**

① 参数 WarnString 是一个字符串变量，它表示了在对话框中要显示的警告信息内容。

② 参数 DlgTitle 是一个字符串变量，它表示对话框的名称。

例如，下述语句可产生出图 5－17 所示的警告信息对话框。

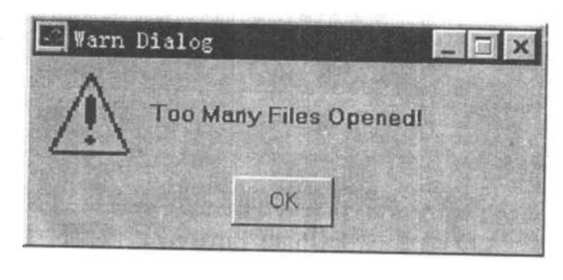


图 5－17　警告信息对话框

```
》Handle = WarnDlg( 'Too Many Files Opened!' , 'Warn Dialog' )
Handle =
    11.0012
```

5.4　MATLAB 下的控件设计技术

用户和 Windows 应用程序之间所进行的信息交换，主要是通过各种控件来实现的。控件不仅是一般窗口中的重要部件，而且也是构成对话框的最主要的部件。在 MATLAB 中，控件

更是系统所定义的10个基本图形对象之一，它有很多属性，以便用户根据需要设计不同种类的控件。

5.4.1 MATLAB的控件种类

在MATLAB中，系统共定义了9种不同类型的控件。这9种控件的外观形式如图5-18所示。

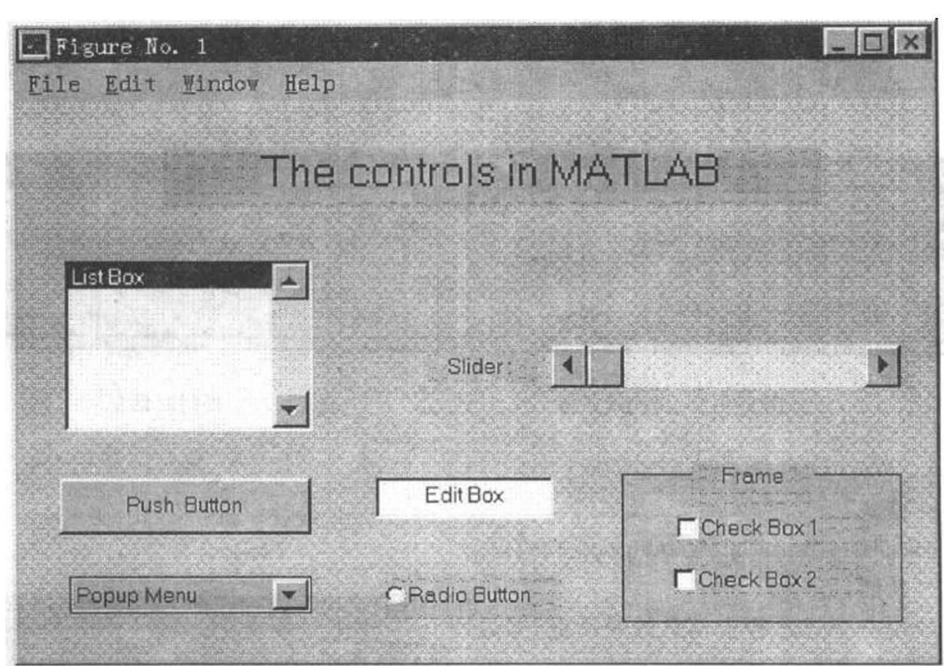


图5-18 MATLAB中的各种控件

MATLAB所使用的这9种控件是：

(1) 命令按钮(Push Button)。该控件主要用于将系统的控制转向某一个程序，以执行某种功能，如"OK(确定)"、"Cancel(取消)"等。

(2) 单选按钮(Radio Button)。该控件允许用户从多种选择项中选择其中的某一项内容，被选中的按钮会在其外圆的中心有一个实心的黑点。这种按钮往往用于让用户选用某种参数或条件。单选按钮只能进行"单项选择"的操作。

(3) 检查框(Cheek Box)。该控件允许用户从多种选择中选择其中的某一些内容，被选中的检查框在其方形外框中有一个对号("√")标记。这种按钮往往用于让用户选用或不选用某种设置内容。检查框可以进行"多项选择"的操作。

(4) 列表框(List Box)。该控件用于从以列表的形式所给出的一些项目中选择出其中的某一项内容，选中的项目将会出现在列表框的最上一行。

(5) 下拉式菜单(Popup Menu)。该控件用于从系统所弹出的由一些命令按钮所组成的菜单中选取某一菜单项并执行相应的功能。

(6) 滑块(Slider)。该控件可以以图示的方式使用户从某一范围的数值中选取某一个数值，该数值的大小是通过滚动杆中滑块的位置来近似显示的。

(7) 编辑框(Edit Box)。该控件是用来向系统输入一些文字或数据的方框,该控件中的文字内容可以由一些编辑键进行操作。

(8) 静态文字(Static Text)。该控件是用来向用户提示某一些信息而显示在窗口中的一些说明文字。

(9) 框架(Frame)。该控件是窗口中的一个矩形方框,方框内的一些控件是一组相关的控件。

5.4.2 控件的实现方法

控件是 MATLAB 所定义的 10 个基本图形对象之一,在某一图形窗口中的各种控件都是由 uicontrol 命令来实现的。

• 格式:

```
hControl = uicontrol( handle , 'PropertyName1' , PropertyValue1 , ...
                       'PropertyName2', PropertyValue2 ,'…')
```

• 功能:

在句柄为 handle 的图形窗口中实现一个控件,并返回该控件的句柄到 hControl 变量。

• 说明:

① 在所实现的控件中,其 PropertyName1 属性的属性值为 PropertyValue1;PropertyName2 属性的属性值为 PropertyValue2……

② 该控件的"双亲"句柄为 handle,若缺省该参数,则其"双亲"为当前图形窗口。

③ 对于该命令所实现的控件,其属性值确定了该控件的外观显示特征及各种特点。

其中重要的属性有"Style"(确定控件的种类形式),"String"(确定控件的表面标记文字),"Value"(控件当前的取值),"Max"(控件能够取的最大值),"Min"(控件能够取的最小值),"Position"(控件的放置位置),"Callback"(选中该控件后所要执行的命令响应序列)等。各个控件的具体属性请参见第 4 章。要注意的是,各个属性参数在不同种类的控件中会有不同的含义。

④ 当某一控件的取值被改变之后,会导致系统去执行由其"Callback"属性所确定的一些命令响应序列。

⑤ 属性"Style"确定了控件的种类形式,它确定了控件的基本外观特征(在下面的各个小节中,我们将对各种形式的控件的实现方法给出示例说明)。Style 属性的取值及其所表示的控件种类见表 5-1。

表 5-1 控件的 Style 属性取值及其对应的控件种类

Style 属性的取值	控 件 种 类	Style 属性的取值	控 件 种 类
pushbutton , push	命令按钮	popupmenu	下拉式菜单
radiobutton , radio	单选按钮	slider	滑块
cheekbox , check	检查框	edit	编辑框
listbox , list	列表框	text	静态文字
		frame	框架

5.4.3 命令按钮的实现方法示例

命令按钮是 Windows 环境中的最基本的控件。下面的示例程序 EX504 说明了命令按钮的实现方法。在该示例程序中，我们定义了 3 个命令按钮，一个用于在图形窗口的绘图区域中画正弦曲线，一个用于画余弦曲线，另一个用于关闭该图形窗口。

```
%**************************************
%    程序：EX504
%    功能：命令按钮实现方法示例
%**************************************
% create a figure window and an axes.
clear
FigWin = figure('Position', [100 100 600 300], ...
               'Name' , 'Uicontrol: Push Button', ...
               'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4  0.2  0.5  0.7],...
                'Box','on');
% create 3 pushbuttons.
push1=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50  200  100  30],...
               'String','Plot sin(x)',...
               'CallBack',...
               ['subplot(AxesHandle);'...
               'x=0:0.1:4*pi;'...
               'plot(x,sin(x));'...
               'axis([ 0  4*pi  -1  1 ]);'...
               'grid on;'...
               'xlabel(''x'');'...
               'ylabel(''y=sin(x)'');']);
   push2=uicontrol(FigWin,...
                 'Style','pushbutton',...
                 'Position',[50  150  100  30],...
                 'String','Plot cos(x)',...
                 'CallBack',...
                 ['subplot(AxesHandle);'...
                 'x=0:0.1:4*pi;'...
                 'plot(x,cos(x));'...
                 'axis([ 0  4*pi  -1  1 ]);'...
                 'grid on;'...
                 'xlabel(''x'');'...
                 'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
```

```
                        'Style','pushbutton',...
                        'Position',[50  100  100  30],...
                        'String','Exit',...
                        'CallBack',['close(FigWin)']);
% draw the initial curve
X = 0:0.1:4 * pi;
plot(X,sin(X));
axis([ 0   4 * pi   -1   1 ]);
xlabel(''x'');
ylabel(''y=sin(x)'');
grid on;
```

图 5 - 19 所示为我们用鼠标点击画正弦曲线的命令按钮之后上述程序所绘制出的对应曲线示例。

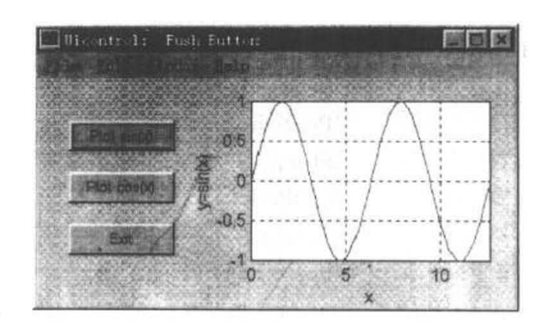


图 5 - 19　命令按钮实现方法示例

5.4.4　单选按钮的实现方法示例

在 Windows 应用程序中，一般是将一组相关的项目设计为一组单选按钮，在操作时，这一组按钮中只能有其中的一个被选中，而其余各按钮都处于未选中的状态，即各个项目之间具有互斥的性质。但是，由于 MATLAB 并未提供一次生成一组单选按钮的功能，它一次只能生成一个按钮，所以这些按钮是可以同时选中而不具备互斥性质的。

要达到各项目间的互斥特性，就要通过程序来实现。这种程序所利用的控件最重要的属性是 Value 属性，当 Value 属性值取 1 时，表明该项目被选中，而当 Value 属性值取 0 时，表明该项目未被选中。最常用的互斥特性程序实现方法是由“If ... Else ... End”语句结构来实现的，这个语句必须出现在每个单选按钮的“Callback”属性中。

例如，示例程序 EX505 说明了单选按钮的这种实现方法。在该示例程序中，我们定义了 3 个单选按钮，它们分别表示画图时所用的颜色：红色、绿色、蓝色，并且是只能取这 3 种颜色之一来画图。当我们选择一个颜色时，图中的曲线就改变为这种颜色。图 5 - 20 为该程序的执行结果示例。

```
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
%        程序：EX505
%        功能：单选按钮实现方法示例
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% create a figure window and an axes.
clear
FigWin = figure('Position', [100 100 600 300], ...
                'Name' , 'Uicontrol: Radio Button', ...
                'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4  0.2  0.5  0.7],...
                'Box','on');
% create 3 radio buttons.
```

```
RadioNum = 3;
for i = 1:RadioNum
  Radio(i)=0;
end
Radio(1)=uicontrol(FigWin,...
                  'Style','radio',...
                  'Position',[50 250 100 30],...
                  'String','Draw in Red',...
                  'CallBack',...
                  ['n=1;'...
                  'if get(Radio(1),''Value'') = =
                  1;'...
                  'set(Radio([1:(n-1),(n+1):RadioNum]),''Value'',0);'...
                  'else '...
                  'set(Radio(1),''Value'',1);'...
                  'end;'...
                  'ColorStr = ''r'';'...
                  'set(FunHandle,''Color'',''red'')']);
Radio(2)=uicontrol(FigWin,...
                  'Style','radio',...
                  'Position',[50  210  100  30],...
                  'String','Draw in Green',...
                  'CallBack',...
                  ['n=2;'...
                  'if get(Radio(2),''Value'')==1;'...
                  'set(Radio([1:(n-1),(n+1):RadioNum]),''Value'',0);'...
                  'else '...
                  'set(Radio(2),''Value'',1);'...
                  'end;'...
                  'ColorStr = ''g'';'...
                  'set(FunHandle,''Color'',''green'')']);
Radio(3)=uicontrol(FigWin,...
                  'Style','radio',...
                  'Position',[50 170 100 30],...
                  'String','Draw in Blue',...
                  'CallBack',...
                  ['n=3;'...
                  'if get(Radio(3),''Value'')==1;'...
                  'set(Radio([1:(n-1),(n+1):RadioNum]),''Value'',0);'...
                  'else '...
                  'set(Radio(3),''Value'',1);'...
                  'end;'...
                  'ColorStr = ''b'';'...
```

图 5-20 单选按钮实现方法示例

```
                'set(FunHandle,''Color'',''blue'')']);
% create 3 push buttons.
push1=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50 120 100 30],...
                'String','Plot sin(x)',...
                'CallBack',...
                ['subplot(AxesHandle);'...
                'X = 0:0.1:4 * pi;'...
                'FunHandle = plot(X,sin(X),ColorStr);'...
                'axis([ 0   4 * pi   -1   1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50 80 100 30],...
                'String','Plot cos(x)',...
                'CallBack',...
                ['subplot(AxesHandle);'...
                'X = 0:0.1:4 * pi;'...
                'FunHandle = plot(X,cos(X),ColorStr);'...
                'axis([ 0   4 * pi   -1   1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50 40 100 30],...
                'String','Exit',...
                'CallBack',['close(FigWin)']);
% draw the initial curve
ColorStr = 'r';
set(Radio(1),'Value',1);
X = 0:0.1:4 * pi;
FunHandle = plot(X,sin(X),ColorStr);
axis([ 0   4 * pi   -1   1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.5 检查框的实现方法示例

检查框类似于一些开关是处于“开”的状态还是处于“关”的状态，从而确定程序是否启用

某些状态。当检查框被选用时，其“Value”属性值为 1；否则，其“Value”属性值为 0。用鼠标单击检查框时，处于被选用状态的检查框会变成非选用状态，处于非选用状态的检查框也会变成被选用状态。

示例程序 EX506 说明了检查框的实现方法。在该示例程序中，我们定义了 3 个检查框，它们分别表示画图时所用的颜色：红色、绿色、蓝色。这里要说明的是，红色、绿色、蓝色这 3 种颜色构成了色彩系统的 3 种基本颜色，称为三原色，我们可以单独使用其中的一种颜色来绘图，也可由几种原色混合成一种颜色来绘图。用三原色的方法表示颜色的形式是矩阵[R G B]，例如[0 0 0] 表示黑色，[1 1 1] 表示白色，[1 0 0] 表示纯红色，[1 1 0] 表示黄色。由于允许同时指定多个基本颜色，所以用 3 个检查框就可以实现这种颜色选用方式。当我们指定一种颜色时，图中的曲线就改变为这种颜色。图 5－21 为该程序的执行结果示例。

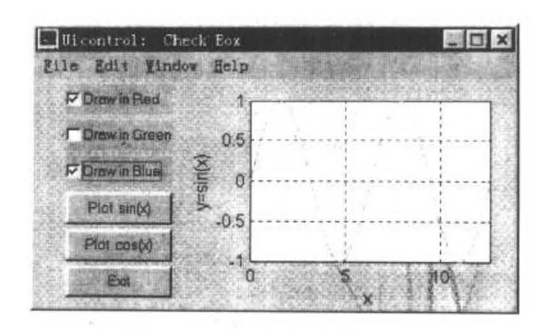


图 5－21　检查框实现方法示例

```
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
%     程序：EX506
%     功能：检查框实现方法示例
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% create a figure window and an axes.
clear
FigWin = figure('Position' , [100 100 600 300], ...
                'Name' , 'Uicontrol: Check Box', ...
                'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4  0.2  0.5  0.7],...
                 'Box','on');
% create 3 checkboxes.
CheckBox1=uicontrol(FigWin,...
                    'Style','checkbox',...
                    'Position',[50  250  100  30],...
                    'String','Draw in Red',...
                    'Value',0,...
                    'CallBack',...
                    ['ColorR = get(CheckBox1,''Value'');'...
                    'ColorG = get(CheckBox2,''Value'');'...
                    'ColorB = get(CheckBox3,''Value'');'...
                    'set(FunHandle,''Color'',[ColorR ColorG ColorB]);']);
CheckBox2=uicontrol(FigWin,...
                    'Style','checkbox',...
                    'Position',[50  210  100  30],...
                    'String','Draw in Green',...
```

```
                    'Value',0,...
                    'CallBack',...
                    ['ColorR = get(CheckBox1,''Value'');'...
                    'ColorG = get(CheckBox2,''Value'');'...
                    'ColorB = get(CheckBox3,''Value'');'...
                    'set(FunHandle,''Color'',[ColorR  ColorG  ColorB]);']);
CheckBox3=uicontrol(FigWin,...
                    'Style','checkbox',...
                    'Position',[50 170 100 30],...
                    'String','Draw in Blue',...
                    'Value',0,...
                    'CallBack',...
                    ['ColorR = get(CheckBox1,''Value'');'...
                    'ColorG = get(CheckBox2,''Value'');'...
                    'ColorB = get(CheckBox3,''Value'');'...
                    'set(FunHandle,''Color'',[ColorR  ColorG  ColorB]);']);
% create 3 push buttons.
push1=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50  120  100  30],...
               'String','Plot sin(x)',...
               'CallBack',...
               ['subplot(AxesHandle);'...
               'X = 0:0.1:4 * pi;'...
               'FunHandle = plot(X,sin(X));'...
               'ColorR = get(CheckBox1,''Value'');'...
               'ColorG = get(CheckBox2,''Value'');'...
               'ColorB = get(CheckBox3,''Value'');'...
               'set(FunHandle,''Color'',[ColorR  ColorG  ColorB]);'...
               'axis([ 0  4*pi  -1  1 ]);'...
               'grid on;'...
               'xlabel(''x'');'...
               'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50  80  100  30],...
               'String','Plot cos(x)',...
               'CallBack',...
               ['subplot(AxesHandle);'...
               'X = 0:0.1:4 * pi;'...
               'FunHandle = plot(X,cos(X));'...
               'ColorR = get(CheckBox1,''Value'');'...
               'ColorG = get(CheckBox2,''Value'');'...
```

```
                'ColorB = get(CheckBox3,''Value'');'...
                'set(FunHandle,''Color'',[ColorR  ColorG  ColorB]);'...
                'axis([ 0  4 * pi  -1  1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50  40  100  30],...
                'String','Exit',... 'CallBack',['close(FigWin)']);
% draw the initial curve
X = 0:0.1:4 * pi;
FunHandle = plot(X,sin(X));
ColorR = get(CheckBox1,'Value');
ColorG = get(CheckBox2,'Value');
ColorB = get(CheckBox3,'Value');
set(FunHandle,'Color',[ColorR  ColorG  ColorB]);
axis([ 0  4 * pi  -1  1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.6 列表框的实现方法示例

列表框中列出了一些可选择的项目，用户可通过鼠标选择其中的一个项目。列表框的这些项目文字内容可由其“String”属性来定义，并且各个项目文字之间用符号“|”来分隔。列表框的“Value”属性值定义了当前所选定的项目号，项目号是指对“String”属性的各个项目内容从1开始向后计数时所对应的项数。

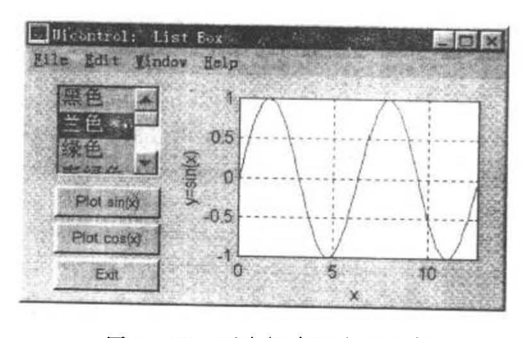


图 5-22　列表框实现方法示例

例如，示例程序 EX507 实现了列表框的功能。在该示例程序中，我们定义了一个含有黑色、蓝色、绿色、青绿色、紫红色、红色、黄色、白色等 8 种颜色的列表框，由该列表框可指定绘图时所用的颜色。图 5-22 为该程序的执行结果示例。

```
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
%     程序：EX507
%     功能：列表框实现方法示例
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% create a figure window and an axes.
clear
```

```
FigWin = figure('Position', [100 100 600 300], ...
              'Name' , 'Uicontrol: List Box', ...
              'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4  0.2  0.5  0.7],...
               'Box','on');
% create a list box.
ListBox=uicontrol(FigWin,...
               'Style','listbox',...
               'Position',[50  180  100  100],...
               'String','黑色|蓝色|绿色|青绿色|紫红色|红色|黄色|白色|',...
               'Valuc',1,...
               'FontSize',12,...
               'CallBack',...
               ['set(FunHandle,''Color'',ColorStr(get(ListBox,''Value'')));']);
% create 3 push buttons.
push1=uicontrol(FigWin,...
              'Style','pushbutton',...
              'Position',[50 120 100 30],...
              'String','Plot sin(x)',...
              'CallBack',...
              ['subplot(AxesHandle);'...
              'X = 0:0.1:4 * pi;'...
              'FunHandle = plot(X,sin(X));'...
              'set(FunHandle,''Color'',ColorStr(get(ListBox,''Value'')));'...
              'axis([ 0  4 * pi  -1  1 ]);'...
              'grid on;'...
              'xlabel(''x'');'...
              'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
              'Style','pushbutton',...
              'Position',[50  80  100  30],...
              'String','Plot cos(x)',...
              'CallBack',...
              ['subplot(AxesHandle);'...
              'X = 0:0.1:4 * pi;'...
              'FunHandle = plot(X,cos(X));'...
              'set(FunHandle,''Color'',ColorStr(get(ListBox,''Value'')));'...
              'axis([ 0  4 * pi  -1  1 ]);'...
              'grid on;'...
              'xlabel(''x'');'...
              'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
              'Style','pushbutton',...
```

```
                    'Position',[50  40  100  30],...
                    'String','Exit',...
                    'CallBack',['close(FigWin)']);
% draw the initial curve
ColorStr = [ 'k','b','g','c','m','r','y','w' ];
X = 0:0.1:4 * pi;
FunHandle = plot(X,sin(X),ColorStr(1));
axis([ 0  4 * pi  -1  1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.7 下拉式菜单的实现方法示例

下拉式菜单也叫组合框。与列表框相类似，下拉式菜单中也列出了一些可供用户选择的项目，用户可通过鼠标选择其中的一个项目。下拉式菜单的这些项目文字内容可由其“String”属性来定义，并且各个项目文字之间用符号“|”来分隔。下拉式菜单的“Value”属性值定义了当前所选定的菜单项目号，项目号是指对“String”属性的各个项目内容从 1 开始向后计数时所对应的项数。下拉式菜单与列表框的区别是前者所占用的区域面积较小。

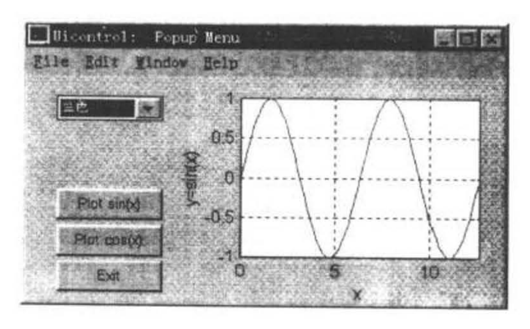


图 5-23　下拉式菜单实现方法示例

例如，示例程序 EX508 实现了下拉式菜单的功能。在该示例程序中，我们定义了一个含有黑色、蓝色、绿色、青绿色、紫红色、红色、黄色、白色等 8 种颜色的下拉式菜单，由该下拉式菜单可指定绘图时所用的颜色。

图 5-23 为该程序的执行结果示例。

```
%*****************************************
%    程序：EX508
%    功能：下拉式菜单实现方法示例
%*****************************************
% create a figure window and an axes.
clear
FigWin = figure('Position', [100 100 600 300], ...
              'Name' , 'Uicontrol: Popup Menu', ... 'NumberTitle', 'off' );
AxesHandle=axes('Position',[0.4  0.2  0.5  0.7],...
              'Box','on');
% create a popup menu.
ComBox=uicontrol(FigWin,...
              'Style','popupmenu',...
              'Position',[50  150  100  100],...
```

```
                'BackGroundColor',[1 1 1],...
                'String','黑色|蓝色|绿色|青绿色|紫红色|红色|黄色|白色|',...
                'Value',1,...
                'CallBack',...
                ['set(FunHandle,''Color'',ColorStr(get(ComBox,''Value'')));']);
% create 3 push buttons.
push1=uicontrol(FigWin,...
            'Style','pushbutton',...
            'Position',[50   120   100   30],...
            'String','Plot sin(x)',...
            'CallBack',...
            ['subplot(AxesHandle);'...
            'X = 0:0.1:4 * pi;'...
            'FunHandle = plot(X,sin(X));'...
            'set(FunHandle,''Color'',ColorStr(get(ComBox,''Value'')));'...
            'axis([ 0   4 * pi   -1   1 ]);'...
            'grid on;'...
            'xlabel(''x'');'...
            'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
            'Style','pushbutton',...
            'Position',[50   80   100   30],...
            'String','Plot cos(x)',...
            'CallBack',...
            ['subplot(AxesHandle);'...
            'X = 0:0.1:4 * pi;'...
            'FunHandle = plot(X,cos(X));'...
            'set(FunHandle,''Color'',ColorStr(get(ComBox,''Value'')));'...
            'axis([ 0   4 * pi   -1   1 ]);'...
            'grid on;'...
            'xlabel(''x'');'...
            'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
            'Style','pushbutton',...
            'Position',[50   40   100   30],...
            'String','Exit',...
            'CallBack',['close(FigWin)']);
% draw the initial curve
ColorStr = [ 'k','b','g','c','m','r','y','w' ];
X = 0:0.1:4 * pi;
FunHandle = plot(X,sin(X),ColorStr(1));
axis([ 0   4 * pi   -1   1 ]);
xlabel('x');
```

```
ylabel('y=sin(x)');
grid on;
```

5.4.8 滑块的实现方法示例

在 Windows 应用程序中，滑块能够以图形示意的方式使用户从某一范围内的数值中选取某一个数值而输入到系统中。

滑块所输入的数值体现在该控件的“Value”属性值上。并且其“Max”属性值是它的“Value”属性所能取的最大值；“Min”属性值是它的“Value”属性所能取的最小值。滑块所体现的数值可由下述 3 种方式进行更改：

(1) 用鼠标拖动滑动块；

(2) 用鼠标单击滑块系统左边或右边的箭头，可使滑块移动 1%；

(3) 用鼠标单击滑块左边或右边的滑动轨道，可使滑块移动 10%。

另外，滑块既可水平放置也可垂直放置。其放置方向由该控件的“Position”属性确定：

(1) 当其“Position”属性的宽度值(第 3 个元素)大于其高度值(第 4 个元素)时，就水平放置滑块；

(2) 当其“Position”属性的宽度值(第 3 个元素)小于其高度值(第 4 个元素)时，就垂直放置滑块。

例如，示例程序 EX509 实现了滑块的功能。在该示例程序中，我们定义了一个能从 1 到 10 变化的滑块位置，由它的取值来指定我们在窗口中画出几个 π 的正弦曲线。图 5-24 为该程序的执行结果示例。

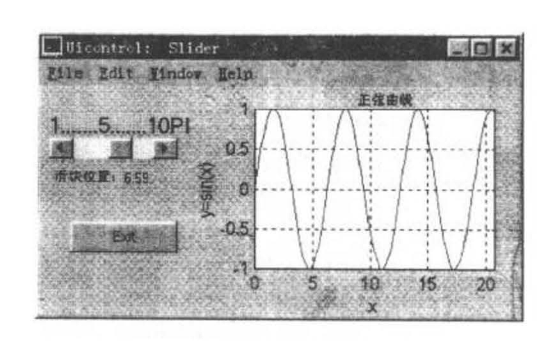


图 5-24 滑块实现方法示例

```
%*****************************************
%    程序：EX509
%    功能：滑块实现方法示例
%*****************************************
% create a figure window and an axes.
clear
FigWin = figure('Position' , [100 100 600 300], ...
                'Name' , 'Uicontrol: Slider', ...
                'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4  0.15  0.5  0.7],...
                'Box','on');
BackColor = get(gcf,'Color');
% create a slider.
SliderBox=uicontrol(FigWin,...
                'Style','slider',...
                'Position',[10  190  170  30],...
```

```
                'Min',1,...
                'Max',10,...
                'Value',2,...
                'BackgroundColor',[ 1   1   1 ],...
                'CallBack',...
                ['Num=get(SliderBox,''Value'');'...
                'set(TextPos,''String'',[''滑块位置:'',num2str(Num)]);'...
                'subplot(AxesHandle);'...
                'x=0:0.1:Num * pi;'...
                'plot(x,sin(x));'...
                'axis([ 0   Num * pi   -1   1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=sin(x)'');']);
% create static text.
TextFlag=uicontrol(FigWin,...
                'Style','text',...
                'Position',[10   230   170   20],...
                'FontSize',12,...
                'BackgroundColor',BackColor,...
                'String','1............5............10');
TextPi = uicontrol(FigWin,...
                'Style','text',...
                'Position',[180   220   20   30],...
                'FontSize',12,...
                'BackgroundColor',BackColor,...
                'String','pi');
TextPos =uicontrol(FigWin,...
                'Style','text',...
                'Position',[15   120   150   40],...
                'FontSize',12,...
                'BackgroundColor',BackColor,...
                'String','滑块位置:2');
TextStr =uicontrol(FigWin,...
                'Style','text',...
                'Position',[330   260   130   30],...
                'FontSize',16,...
                'BackgroundColor',BackColor,...
                'String','正弦曲线');
% create exit pushbutton.
ExitButton=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50   80   100   30],...
```

```
                'String','Exit',...
                'CallBack',['close(FigWin)']);
% draw the initial curve
Num = get(SliderBox,'Value');
X = 0:0.1:Num * pi;
plot(X,sin(X));
axis([ 0   Num * pi   -1   1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.9 静态文字的实现方法示例

静态文字主要用于在窗口中显示一些提示性的文字说明信息，以方便用户在当前状态下的操作，从而使得用户可以脱离操作手册而进行程序操作。静态文字的内容体现在该控件的“String”属性上，位置由“Position”属性来指定。

要说明的是，这里所使用的静态文字中“静态”二字的含义主要是指用户不能更改这些文字的显示内容。但是，当程序运行时，却可以由程序更改其“String”属性值，从而达到不同情况下显示不同提示信息的目的。

例如，示例程序 EX510 实现了静态文字的修改功能。在该示例程序中，根据所画出的曲线类型，自动确定图形的标题是“正弦曲线”或“余弦曲线”。

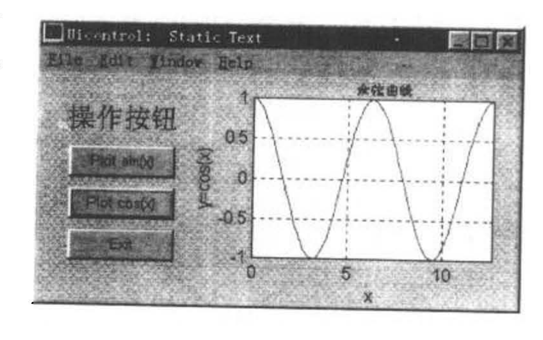


图 5-25 静态文字实现方法示例

图 5-25 为该程序的执行结果示例。

```
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
%      程序：EX510
%      功能：静态文字实现方法示例
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% create a figure window and an axes.
clear
FigWin = figure('Position' , [100 100 600 300], ...
               'Name' , 'Uicontrol: Static Text', ...
               'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4   0.15   0.5   0.7],...
                'Box','on');
BackColor = get(gcf,'Color');
% create static text.
Text1=uicontrol(FigWin,...
               'Style','text',...
               'Position',[35   230   130   30],...
```

```
                    'FontSize',16,...
                    'BackgroundColor',BackColor,...
                    'String','操作按钮');
Text2=uicontrol(FigWin,...
                    'Style','text',...
                    'Position',[330  260  130  30],...
                    'FontSize',16,...
                    'BackgroundColor',BackColor,...
                    'String','正弦曲线');
% create 3 pushbuttons.
push1=uicontrol(FigWin,...
                    'Style','pushbutton',...
                    'Position',[50  180  100  30],...
                    'String','Plot sin(x)',...
                    'CallBack',...
                    ['set(Text2,''String'',''正弦曲线'');'...
                    'subplot(AxesHandle);'...
                    'x=0:0.1:4*pi;'...
                    'plot(x,sin(x));'...
                    'axis([ 0  4*pi  -1  1 ]);'...
                    'grid on;'...
                    'xlabel(''x'');'...
                    'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
                    'Style','pushbutton',...
                    'Position',[50  130  100  30],...
                    'String','Plot cos(x)',...
                    'CallBack',...
                    ['set(Text2,''String'',''余弦曲线'');'...
                    'subplot(AxesHandle);'...
                    'x=0:0.1:4*pi;'...
                    'plot(x,cos(x));'...
                    'axis([ 0  4*pi  -1  1 ]);'...
                    'grid on;'...
                    'xlabel(''x'');'...
                    'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
                    'Style','pushbutton',...
                    'Position',[50  80  100  30],...
                    'String','Exit',...
                    'CallBack',['close(FigWin)']);
% draw the initial curve
X = 0:0.1:4*pi;
```

```
plot(X,sin(X));
axis([ 0   4 * pi   -1   1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.10 编辑框的实现方法示例

编辑框可供用户输入某一串文字。对这一串文字，用户还可利用一些编辑键来进行修改操作。

用户在编辑框中所输入的文字内容体现在该控件的"String"属性上，编辑框的位置由"Position"属性来指定。

例如，示例程序 EX511 实现编辑框的功能。在该示例程序中，我们定义了一个能接受用户输入数据的编辑框，编辑框中的数据指定了我们要在绘图窗口中画出几个 π 的正弦曲线。

图 5－26 为该程序的执行结果示例。

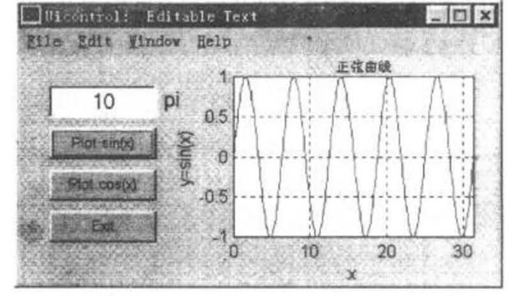


图 5－26 编辑框实现方法示例

```
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
%      程序：EX511
%      功能：编辑框实现方法示例
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
% create a figure window and an axes.
clear
FigWin = figure('Position' , [100 100 600 300], ...
               'Name' , 'Uicontrol: Editable Text', ...
               'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4   0.15   0.5   0.7],...
                'Box','on');
BackColor = get(gcf,'Color');
% create editable text.
EditBox=uicontrol(FigWin,...
                'Style','edit',...
                'Position',[50   230   100   30],...
                'FontSize',12,...
                'String','4',...
                'BackgroundColor',[ 1   1   1 ],...
                'CallBack',...
                ['NumStr=get(EditBox,''String'');'...
                'Num=str2num(NumStr);'...
                'subplot(AxesHandle);'...
                'x=0:0.1:Num * pi;'...
```

```
                'if SinFlag==1 plot(x,sin(x)); else plot(x,cos(x)); end;'...
                'axis([ 0   Num * pi   -1   1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=sin(x)'');']);
% create static text.
TextPi =uicontrol(FigWin,...
                'Style','text',...
                'Position',[160   230   20   30],...
                'FontSize',12,...
                'BackgroundColor',BackColor,...
                'String','pi');
TextStr=uicontrol(FigWin,...
                'Style','text',...
                'Position',[330   260   130   30],...
                'FontSize',16,...
                'BackgroundColor',BackColor,...
                'String','正弦曲线');
% create 3 pushbuttons.
push1=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50   180   100   30],...
               'String','Plot sin(x)',...
               'CallBack',...
               ['set(TextStr,''String'',''正弦曲线'');'...
               'SinFlag=1;'...
               'NumStr=get(EditBox,''String'');'...
               'Num=str2num(NumStr);'...
               'subplot(AxesHandle);'...
               'x=0:0.1:Num * pi;'...
               'plot(x,sin(x));'...
               'axis([ 0   Num * pi   -1   1 ]);'...
               'grid on;'...
               'xlabel(''x'');'...
               'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50   130   100   30],...
               'String','Plot cos(x)',...
               'CallBack',...
               ['set(TextStr,''String'',''余弦曲线'');'...
               'SinFlag=0;'...
               'NumStr=get(EditBox,''String'');'...
```

```
                'Num=str2num(NumStr);'...
                'subplot(AxesHandle);'...
                'x=0:0.1:Num * pi;'...
                'plot(x,cos(x));'...
                'axis([ 0   Num * pi   -1   1 ]);'...
                'grid on;'...
                'xlabel(''x'');'...
                'ylabel(''y=sin(x)'');']);
push3=uicontrol(FigWin,...
                'Style','pushbutton',...
                'Position',[50   80   100   30],...
                'String','Exit',...
                'CallBack',['close(FigWin)']);
% draw the initial curve
SinFlag=1;
X = 0:0.1:4 * pi;
plot(X,sin(X));
axis([ 0   4 * pi   -1   1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.4.11 框架的实现方法示例

引入框架的目的是为了将窗口中某些相似功能的控件构成一个群组，在这些控件周围所设置的一个矩形方框就是框架。框架控件的设置，使得人们便于理解哪些控件是一个群体，哪些控件有是另一组群体。每组控件能完成一些相关的功能，这样就可以提高窗口的视觉理解效果。

例如，示例程序 EX512 说明了框架的实现方法及其应用。在该示例程序中，我们把 3 个命令按钮用一个框架包括了起来。

图 5-27 为该程序的执行结果示例。

```
%****************************************
%     程序：EX512
%     功能：框架实现方法示例
%****************************************
% create a figure window and an axes.
clear
FigWin = figure('Position' , [100   100   600   300], ...
               'Name' , 'Uicontrol: Frames', ...
               'NumberTitle' , 'off' );
AxesHandle=axes('Position',[0.4   0.15   0.5   0.7],...
                'Box','on');
```

```
BackColor = get(gcf,'Color');
% create a frame.
FrameBox=uicontrol(FigWin,...
                   'Style','frame',...
                   'Position',[30 50 140 230]);
% create static text.
Text1=uicontrol(FigWin,...
               'Style','text',...
               'Position',[35  230  130  30],...
               'FontSize',16,...
               'String','操作按钮');
Text2=uicontrol(FigWin,...
               'Style','text',...
               'Position',[330  260  130  30],...
               'FontSize',16,...
               'BackgroundColor',BackColor,...
               'String','正弦曲线');
% create 3 pushbuttons.
push1=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50  180  100  30],...
               'String','Plot sin(x)',...
               'CallBack',...
               ['set(Text2,''String'',''正弦曲线'');'...
               'subplot(AxesHandle);'...
               'x=0:0.1:4*pi;'...
               'plot(x,sin(x));'...
               'axis([ 0  4*pi  -1  1 ]);'...
               'grid on;'...
               'xlabel(''x'');'...
               'ylabel(''y=sin(x)'');']);
push2=uicontrol(FigWin,...
               'Style','pushbutton',...
               'Position',[50  130  100  30],...
               'String','Plot cos(x)',...
               'CallBack',...
               ['set(Text2,''String'',''余弦曲线'');'...
               'subplot(AxesHandle);'...
               'x=0:0.1:4*pi;'...
               'plot(x,cos(x));'...
               'axis([ 0  4*pi  -1  1 ]);'...
               'grid on;'...
               'xlabel(''x'');'...
```

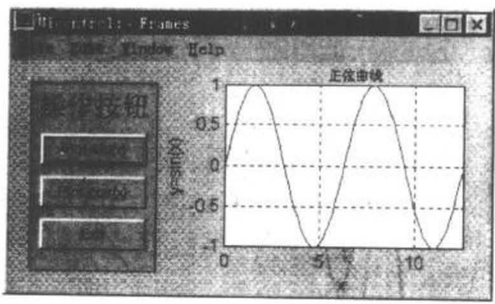


图 5-27 框架实现方法示例

```
                 'ylabel(''y=cos(x)'');']);
push3=uicontrol(FigWin,...
                 'Style','pushbutton',...
                 'Position',[50   80   100   30],...
                 'String','Exit',...
                 'CallBack',['close(FigWin)']);
% draw the initial curve
X = 0:0.1:4 * pi;
plot(X,sin(X));
axis([ 0   4 * pi   -1   1 ]);
xlabel('x');
ylabel('y=sin(x)');
grid on;
```

5.5 MATLAB 的可视化 GUI 设计工具 Guide 简介

通过前面一节的编程实例，我们可以体会到 GUI 的设计并不是一件很简单的事，相反，它还是一件比较繁杂的工作。在 GUI 的设计过程中，我们不仅要仔细地计算每个控件的放置位置，还要正确地确定其各种属性值，整个过程还需要不断地进行返工，才有可能设计出满意的 GUI。另外，在编写回调函数时，其书写格式较为复杂，稍不小心就很容易出错。顺应可视化编程的潮流，MATLAB 5.X 也提供了一个可视化的 GUI 设计工具 —— Guide。本节对这个工具作简要介绍。

5.5.1 Guide 的组成部分

在 MATLAB 的命令窗口中键入命令"guide"，即可调用这个 GUI 设计工具。启用这个 GUI 设计工具后的画面如图 5-28 所示。

由图 5-28 可以看出，Guide 是由下述 5 个工具组成的：

(1) 菜单编辑器(Menu Editor)；

(2) 控件工具板(New Object Palette)；

(3) 控件对齐工具(Alignment Tool)；

(4) 属性编辑器(Property Editor)；

(5) 回调函数编辑器(Callback Editor)。

用鼠标单击图 5-28 中的各个图标命令按钮，即可调用相应的工具。

5.5.2 菜单编辑器

单击"Menu Editor"按钮，即可调用图 5-29 所示的菜单编辑器。

在该菜单编辑器中，我们可以设计菜单的结构，输入菜单项目的名称，指定各个菜单命令的回调函数名称等，如图中的"uimenu1"，"uimenu2"，"uimenu3"，"Subuimenu1" "Subuimenu2"。

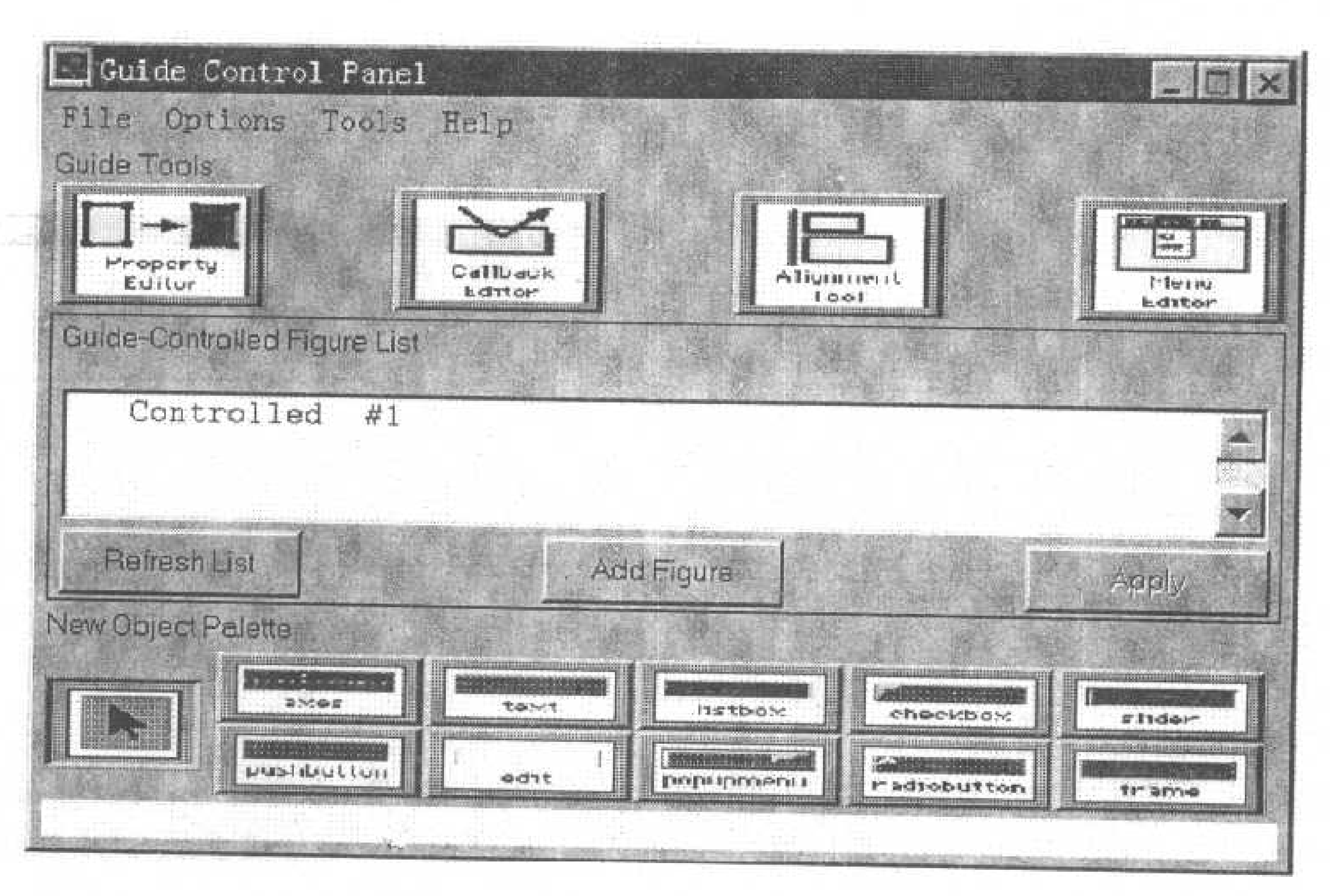


图 5 - 28 GUI 编程设计工具 Guide

图 5 - 29 菜单编辑器

5.5.3 控件工具板

图 5-28 下半部的 10 个图标命令按钮组成了 MATLAB 的控件工具板。单击其中的一个图标，即可调用对应的控件工具，然后我们就可用鼠标拖动的方式将该控件放置到图形窗口中。如图 5-30 所示为控件的设计过程。

利用这种方法，我们可以在图形窗口中直接用鼠标拖动的方式放置各种控件，而不用再去精确计算各个控件的安放位置，不用再去书写其控件类型，只须进行一些鼠标拖动操作，就可完成 GUI 画面的设计，十分便利。

图 5-30 控件设计过程示意

5.5.4 控件对齐工具

单击"Alignment Tool"按钮，即可调用图 5-31 所示的控件对齐工具。

利用这个控件对齐工具，我们可以进行多个控件的水平对齐、垂直对齐，可以分布各个控件在图形窗口中的位置，也可以设计控件之间的间距，以便设计出更加美观的 GUI 画面。

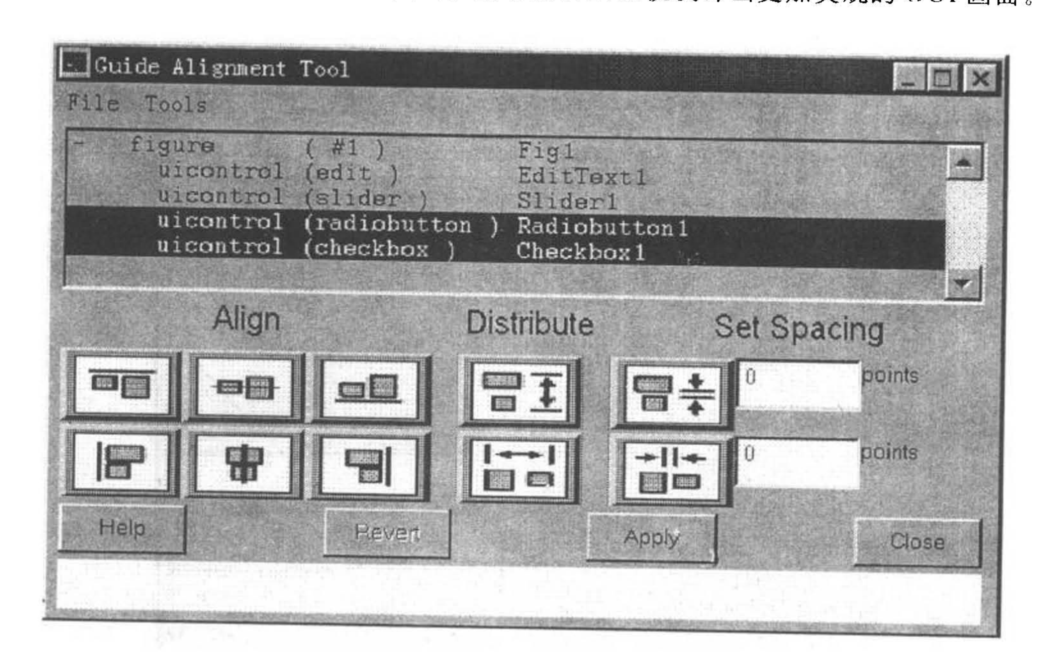


图 5-31 控件对齐工具

5.5.5 属性编辑器

单击"Property Editor"按钮，即可调用图 5-32 所示的属性编辑器。

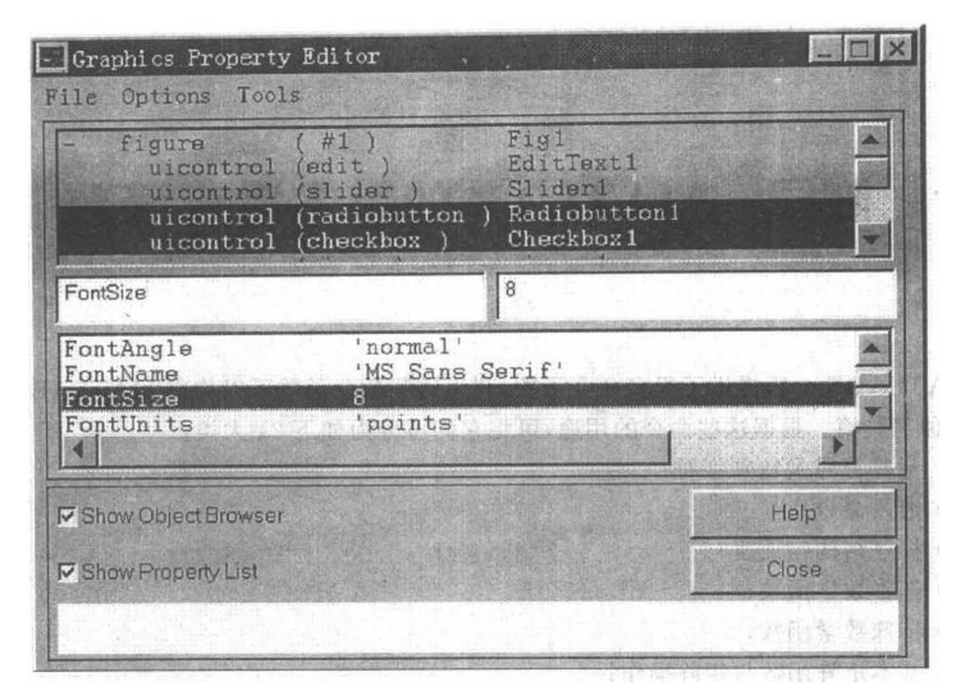


图 5-32 属性编辑器

利用这个属性编辑器,我们可以选定控件的某个属性,然后指定或输入其属性值,以便设计出不同外观特征的控件。

5.5.6 回调函数编辑器

单击"Callback Editor"按钮,即可调用图 5-33 所示的回调函数编辑器。

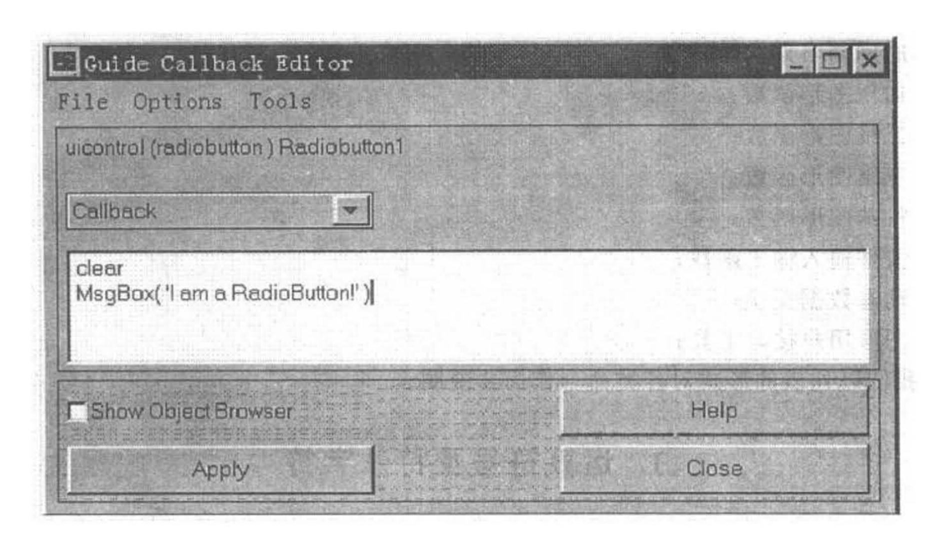


图 5-33 回调函数编辑器

利用这个回调函数编辑器,我们可以像编写一般的 MATLAB 程序一样来编写回调函数,而不必过多地去考虑回调函数的书写格式。

附录 MATLAB 的基本命令函数

MATLAB 5.1 版提供了很多命令函数，利用这些命令函数可以构成 MATLAB 的各种程序及各种工具箱。根据这些命令的用途，可将它们可分为如下 21 大类：

(1) 运算符号及特殊字符；
(2) 通用命令；
(3) 程序结构与语言；
(4) 基本数学函数；
(5) 特殊数学函数；
(6) 基本矩阵函数与矩阵操作；
(7) 稀疏矩阵；
(8) 矩阵函数及线性代数；
(9) 数据类型变换与数据结构；
(10) 数据分析与傅里叶变换函数；
(11) 泛函及 ODE 求解程序；
(12) 插值运算与多项式；
(13) 字符串处理函数；
(14) 时间与日期函数；
(15) 通用图形函数；
(16) 二维图形函数；
(17) 三维图形函数；
(18) 特殊图形函数；
(19) 文件输入输出函数；
(20) 动态数据交换；
(21) 图形用户接口工具；

下面我们以列表的形式对各类命令作简要说明。

1. 运算符号及特殊字符

算术运算函数与算术运算符		
plus	+	加法
uplus	+	一元加法

续表

minus	-	减法
uminus	-	一元减法
mtimes	*	矩阵乘法
times	.*	数组乘法
mpower	^	矩阵乘方
power	.^	数组乘方
mldivide	\	反斜线符号或矩阵左除
mrdivide	/	斜线符号或矩阵右除
ldivide	.\	数组左除
rdivide	./	数组右除
kron	kron	Kronecker 张量积
	关系运算符	
eq	==	等于
ne	~=	不等于
lt	<	小于
gt	>	大于
le	<=	小于或等于
ge	>=	大于或等于
	逻辑运算符	
and	&	逻辑与
or	|	逻辑或
not	~	逻辑非
xor		逻辑异或
any		向量任一元素非零则返回真
all		向量所有元素非零则返回真
	特殊字符	
colon	:	冒号
paren	( )	圆括号及下标
paren	[ ]	方括号
paren	{ }	花括号及下标
punct	.	小数点
punct	.	结构域访问符
punct	..	父目录
punct	...	续行符
punct	,	分隔符
punct	;	分号
punct	%	注释符
punct	!	调用操作系统命令
punct	=	赋值号
punct	'	引用符号
transpose	.'	转置运算
ctranspose	'	复数转置运算
horzcat	[,]	水平联结符

续表

vertcat	[;]	垂直联结符
subsasgn	( ), { }, .	下标分配
subsref	( ), { }, .	下标参考
subsindex		下标索引

位操作符

bitand	位与
bitcmp	位补
bitor	位或
bitmax	最大浮点整数
bitxor	位异或
bitset	位设置
bitget	位读取
bitshift	位移位

集合运算符

union	集合联合运算
unique	集合元素惟一化运算
intersect	集合求交运算
setdiff	集合求差运算
setxor	集合异或运算
ismember	是集合的元素时返回真

2. 通用命令

通用信息

help	在线帮助
helpwin	在线帮助窗口
helpdesk	在线帮助工作台
demo	运行演示程序
ver	版本信息
whatsnew	显示 MATLAB 5.1 的新功能
readme	显示 Readme 文件

工作空间管理

who	列当前变量
whos	列当前变量详细信息
clear	清除内存中的变量和函数
pack	整理工作空间内存
load	从磁盘文件调入变量到工作空间
save	将工作空间的变量存入到磁盘文件
quit	退出 MATLAB

命令管理及函数管理

what	列出指定目录的 MATLAB 指定文件

续表

type	列出 M 文件
edit	编辑 M 文件
lookfor	在 Help 信息中搜索关键字
which	定位函数及文件
pcode	产生预解析的伪代码文件(P 文件)
inmem	列出内存中的函数
mex	编译 MEX 函数
搜索路径管理	
path	获取或设置搜索路径
addpath	给搜索路径添加目录名
rmpath	从搜索路径中去除目录名
editpath	编辑搜索路径
命令窗口控制	
echo	命令回显
more	命令窗口的分页输出控制
diary	保存 MATLAB 的运行过程
format	设置输出格式
操作系统命令	
cd	改变当前工作目录
pwd	显示当前工作目录
dir	列出目录内容
delete	删除文件
getenv	获取环境变量
!	执行操作系统命令
dos	执行 DOS 命令
unix	执行 UNIX 命令
vms	执行 VMSDCL 命令
web	打开 Web 浏览器
computer	显示计算机类型
程序调试命令	
debug	列出调试命令
dbstop	设置程序运行断点
dbclear	去除程序断点
dbcont	继续执行
dbdown	改变局部工作空间内容
dbstack	显示函数调用堆栈
dbstatus	列出所有程序断点
dbstep	执行一条或多条命令
dbtype	列出带有行号的 M 文件
dbup	改变局部工作空间内容
dbquit	退出程序调试模式
dbmex	调试 MEX 文件(仅用于 UNIX 系统)

续表

M 文件状况	
profile	M 文件的执行时间
旧版本函数	
mexdebug	调试 MEX 文件
其　他	
binpatch	修补二进制文件
doc	装载 HTML 文档到 Web 浏览器的工具
exit	退出 MATLAB 系统
helpinfo	帮助选项的信息
info	MATLAB 及 MathWorks 公司的信息
isstudent	是 MATLAB 的学生用版本时返回真
isunix	是 MATLAB 的 UNIX 版本时返回真
isvms	是 MATLAB 的 VMS 版本时返回真
isppc	是 MATLAB 的 Macintosh Power PC 版本时返回真
isieee	是符合 IEEE 结构计算机时返回真
ls	列出目录内容
memory	内存空间大小的帮助信息
notebook	打开 Word 环境中的一个 m 书籍
nnload	装载 Netscape 站点导航器
profsumm	汇总状况信息
subscribe	订阅 MathWorks 新闻简报
whichcls	返回输入类型
GUI(图形用户接口)工具	
editarray	以图形的方式编辑一个数组
maeasgn	将表达式的值赋给数组的一部分
maeresize	改变矩阵的维数分配情况
mauifindexe	返回可执行的 MAUI 的绝对路径
mdbstatus	调试器/编辑器的数据状态
miport	返回 MATLAB 的端口号
genpath	生成基于工具箱的合理路径
path2rc	将 MATLAB 的当前路径存储到“pathdef. m”文件
pathtool	Macintosh 机和 PC 机的路径浏览器
workspace	Macintosh 机和 PC 机的工作空间浏览器

3. 程序结构与语言

程序控制流程	
if	条件执行语句
else	if 语句的条件说明
elseif	ih 语句的条件说明

续表

end	if,for,while,switch 语句的结束说明
for	for 循环语句
while	while 循环语句
break	从 for 或 while 循环体中跳出
switch	switch 开关条件语句
case	开关条件说明语句
otherwise	缺省开关条件说明语句
return	返回调用函数
赋值与执行	
eval	执行由字符串所构成的 MATLAB 命令
feval	执行由字符串所构成的函数
evalin	执行工作空间的表达式
builtin	执行内建函数
assignin	给工作空间中的变量赋值
run	运行主程序
程序、函数与变量	
script	MATLAB 程序的说明
function	定义一个新函数
global	定义全局变量
mfilename	返回当前执行函数的名称
lists	逗号分隔表
exist	检查变量或函数是否已经定义
isglobal	是全局变量时返回真
程序参数处理	
nargchk	有效输入参数的个数
nargin	函数输入参数的个数
nargout	函数输出参数的个数
varargin	输入参数的变量长度
varargout	输出参数的变量长度
inputname	输入参数名称
信息显示	
error	显示错误信息并终止函数运行过程
warning	显示警告信息
lasterr	上一个错误信息
errortrap	在测试时跳过错误
disp	显示数组
fprintf	显示格式化的信息
sprintf	将格式化的信息写入到一个字符串
交互输入	
input	提示用户输入
keyboard	键盘输入
pause	暂停
uimenu	生成用户接口菜单
uicontrol	生成用户接口控件

4. 基本数学函数

三角函数	
sin	正弦函数
sinh	双曲正弦函数
asin	反正弦函数
asinh	反双曲余弦函数
cos	余弦函数
cosh	双曲余弦函数
acos	反余弦函数
acosh	反双曲余弦函数
tan	正切函数
tanh	双曲正切函数
atan	反正切函数
atan2	第四象限反正切函数
atanh	反双曲正切函数
sec	正割函数
sech	双曲正割函数
asec	反正割函数
asech	反双曲正割函数
csc	余割函数
csch	双曲余割函数
acsc	反余割函数
acsch	反双曲余割函数
cot	余切函数
coth	双曲余切函数
acot	反余切函数
acoth	反双曲余切函数
指数函数	
exp	指数函数
log	自然对数函数
log10	常用对数函数
log2	以 2 为底的对数函数
pow2	以 2 为底的幂函数
sqrt	平方根函数
nextpow2	nextpow2(N)返回满足 2^P>=abs(N)的最小的 P 值
复数函数	
abs	绝对值函数
angle	相位角
conj	复共轭
imag	负数虚部
real	负数实部
unwrap	展开相位角

续 表

isreal	是实数数组时返回真
cplxpair	将数字分类为复共轭对
数值处理函数	
fix	朝零方向取整
floor	朝－∞方向取整
ceil	朝＋∞方向取整
round	朝最近的整数取整
mod	模运算
rem	除后取余
sign	符号函数

5. 特殊数学函数

特殊数学函数	
airy	Airy 函数
besselj	第一类 Bessel(贝塞尔)函数
bessely	第二类 Bessel(贝塞尔)函数
besselh	第三类 Bessel(贝塞尔)函数(Hankel 函数)
besseli	改进的第一类 Bessel(贝塞尔)函数
besselk	改进的第二类 Bessel(贝塞尔)函数
beta	β类函数
betainc	非完全的 β 函数
betaln	β 函数的对数
ellipj	Jacobi(雅可比)椭圆函数
ellipke	完全椭圆积分
erf	误差函数
erfc	互补误差函数
erfcx	比例互补误差函数
erfinv	逆误差函数
expint	指数积分函数
gamma	γ 函数
gammainc	非完全 γ 函数
gammaln	γ 函数的对数
legendre	关联 Legendre 函数
cross	向量叉积
数字理论函数	
factor	素数
isprime	是素数时返回真
primes	生成一列素数
gcd	最大公约数
lcm	最小公倍数

续表

rat	有理逼近
rats	有理输出
perms	所有的排列
nchoosek	从 N 个元素中一次取出 K 个元素的所有组合
坐标变换	
cart2sph	将笛卡尔坐标变换为球坐标
cart2pol	将笛卡尔坐标变换为极坐标
pol2cart	将极坐标变换为笛卡尔坐标
sph2cart	将球坐标变换为笛卡尔坐标
hsv2rgb	将色彩饱和值变换为 RGB 向量
rgb2hsv	将 RGB 向量变换为色彩饱和值
其他函数	
dot	向量点积
besschk	检查 Bessel 函数的变元
bessela	原有的 Bessel 函数
betacore	不完全 β 函数的核心算法
erfcore	误差函数的核心算法

6. 基本矩阵函数与矩阵操作

基本矩阵	
zeros	生成零矩阵
ones	生成全 1 矩阵
eye	生成单位矩阵
repmat	矩阵块复制
rand	生成均匀分布的随机矩阵
randn	生成正态分布的随机矩阵
linspace	生成线性间隔的向量
logspace	生成对数间隔的向量
meshgrid	生成绘制三维图形时的 X 与 Y 数组
:	生成规则间隔的向量
基本矩阵信息	
size	矩阵尺寸函数
length	向量长度函数
ndims	矩阵维数
disp	显示矩阵或字符串
isempty	是空矩阵时返回真
isequal	两矩阵相同时返回真
isnumeric	是数值矩阵时返回真
islogical	是逻辑矩阵时返回真
logical	将数值矩阵变换为逻辑矩阵

续表

矩阵操作	
reshape	改变矩阵大小
diag	建立或提取对角矩阵
tril	取矩阵的下三角部分
triu	取矩阵的上三角部分
fliplr	左右旋转矩阵
flipud	上下旋转矩阵
flipdim	沿指定的维数翻转矩阵
rot90	矩阵旋转 90°
:	矩阵的索引号,重新排列矩阵
find	建立非零元素的索引
end	最后一个索引
sub2ind	由下标变换为线性索引
ind2sub	由线性索引变换为下标
特殊变量和常数	
ans	当前的答案
eps	浮点数的精度
realmax	最大浮点数
realmin	最小浮点数
pi	圆周率值(pi=3.1415926535897…)
i,j	虚数单位
inf	无穷大
NaN	非数值(Not a Number)
isnan	当不是一个数值时返回真
isinf	是无穷大时返回真
isfinite	不是无穷大时返回真
flops	浮点运算次数
why	对问题的一个简明的答案
特殊矩阵	
compan	伴随矩阵
gallery	几个小的测试矩阵
hadamard	Hadamard 矩阵
hankel	Hankel 矩阵
hilb	Hilbert 矩阵
invhilb	Hilbert 逆矩阵
magic	魔方矩阵
pascal	Pascal 矩阵
rosser	经典的对称特征值测试问题
toeplitz	Toeplitz 矩阵
vander	Vandermonde 矩阵
wilkinson	Wilkinson 特征值测试

7. 稀疏矩阵

基本稀疏矩阵	
speye	稀疏单位矩阵
sprand	稀疏均匀分布的随机矩阵
sprandn	稀疏随机矩阵
sprandsym	对称的稀疏随机矩阵
spdiags	从对角矩阵中形成稀疏矩阵
完全矩阵与稀疏矩阵之间的变换	
sparse	建立稀疏矩阵
full	将稀疏矩阵变为完全矩阵
find	寻找非零元素的序号
spconvert	稀疏矩阵外部结构的变换
稀疏矩阵的操作	
nnz	非零元素的数目
nonzeros	非零矩阵元素
nzmax	分配给非零元素的存储空间
spones	用“1”取代非零元素
spalloc	为非零元素分配内存
issparse	当矩阵为稀疏矩阵时其值为真
spfun	只对非零元素取函数
spy	显示稀疏矩阵
排序算法	
colmmd	列最小度
symmmd	最小对称度
symrcm	逆 Cuthill-McKee 序
colperm	基于非零元素按列排列
randperm	随机向量排列
dmperm	Dulmage-Mendelsohn 分解
线性代数	
eigs	求矩阵的特征值与特征向量
svds	矩阵的奇异值分解(SVD 分解)
luinc	矩阵的不完全三角分解(LU 分解)
cholinc	矩阵的不完全 Cholesky 分解
normest	估计 $\|\cdot\|_2$ 范数
condest	估计 $\|\cdot\|_1$ 范数
sprank	结构化秩
线性方程迭代方法	
pcg	预处理的共轭梯度方法
bicg	双共轭梯度方法
bicgstab	双共轭梯度稳定方法
cgs	共轭梯度平方方法
gmres	通用最小余值方法
qmr	准最小余值方法

续表

树的操作方法	
treelayout	显示一个或多个树结构
treeplot	画树结构
etree	求矩阵的消元树
etreeplot	画消元树
gplot	画出一个树
混合处理方法	
symbfact	符号分解分析
spparms	为稀疏矩阵处理过程设置参数
spaugment	形成最小二乘增广系统
工具函数	
rjr	随机 Jacobi 旋转
sparsfun	稀疏辅助函数与参数
旧版本函数	
unmesh	把一些边转换成图或矩阵

8. 矩阵函数及线性代数

矩阵分析	
norm	矩阵或向量的范数
normest	估计矩阵的 2-范数
rank	矩阵的秩
det	矩阵行列式的值
trace	矩阵的迹
null	零矩阵
orth	矩阵正交化
rref	减缩行格式矩阵
subspace	子空间
线性方程	
\ 及 /	线性方程求解
inv	矩阵求逆
cond	矩阵的条件数
condest	估计 $\|\cdot\|_1$ 范数
chol	Cholesky 分解
cholinc	不完全 Cholesky 分解
lu	LU 分解
luinc	不完全 LU 分解
qr	正交三角矩阵分解
nnls	非负最小二乘
pinv	矩阵伪逆
lscov	协方差已知的情况下最小二乘求解

续表

特征值与奇异值	
eig	求特征值和特征向量
svd	奇异值分解
eigs	求一组特征值
svds	求一组奇异值
poly	求特征多项式
polyeig	多项式特征值问题
condeig	求关于特征值的条件数
hess	Hessenberg 形式
qz	求广义特征值
schur	Schur 分解
矩阵函数	
expm	矩阵指数
logm	矩阵对数
sqrtm	矩阵开平方根
funm	求一般矩阵函数的值
分解工具	
qrdelete	从 QR 分解中消去一列
qrinsert	从 QR 分解中插入一列
rsf2csf	将实分块对角矩阵变为复对角形式
cdf2rdf	将复对角矩阵变为实分块对角形式
balance	将矩阵平衡处理以提高特征值精度
planerot	Given 平面旋转
其他共享函数	
expm1	通过 Pade 近似算法求解矩阵指数
expm2	通过 Taylor 级数求矩阵指数
expm3	通过矩阵特征值和特征向量求矩阵指数
旧版本函数	
rcond	LINPACK 逆条件值估计

9. 数据类型变换与数据结构

数据类型变换	
double	变换为双精度
sparse	生成稀疏矩阵
char	生成字符数组(字符串)
cell	生成单元数组
struct	生成或变换为结构数组
uint8	变换为无符号 8 位整数
inline	构造 INLINE 对象

续表

多维数组函数	
cat	连接多个数组
ndims	数组维数
ndgrid	生成用于 N-D 函数和插值的数组
permute	改变数组维数的序列
ipermute	倒转改变数组维数的序列
shiftdim	维数移位
squeeze	除掉单独维
单元数组函数	
cell	生成单元数组
celldisp	显示单元数组内容
cellplot	显示单元数组的图形描述
num2cell	将数值数组转变为单元数组
deal	将输入分配至输出
cell2struct	将单元数组转变为结构数组
struct2cell	将结构数组转变为单元数组
iscell	是单元数组时返回真
数据结构函数	
struct	生成或转变为结构数组
fieldnames	获取数据结构的域名
getfield	获取数据结构的域内容
setfield	设置数据结构的域内容
rmfield	删除数据结构的域
isfield	如果域名是结构数组的域名时返回真
isstruct	是结构时返回真
面向对象的编程函数	
class	生成一个对象或返回一个对象的类
struct	将对象转变为一个结构数组
methods	显示类方法名称
isa	如果对象是给定的类则返回真
isobject	是对象时返回真
inferiorto	降低类关系
superiorto	提高类关系
可超负荷计算算子	
minus	用于计算 a-b 的可超负荷算子
plus	用于计算 a+b 的可超负荷算子
times	用于计算 a. * b 的可超负荷算子
mtimes	用于计算 a * b 的可超负荷算子
mldivide	用于计算 a\b 的可超负荷算子
mrdivide	用于计算 a/b 的可超负荷算子
rdivide	用于计算 a. /b 的可超负荷算子
ldivide	用于计算 a. \b 的可超负荷算子
power	用于计算 a. ^ b 的可超负荷算子

续表

mpower	用于计算 a^ b的可超负荷算子
uminus	用于计算—a 的可超负荷算子
uplus	用于计算+a 的可超负荷算子
horzcat	用于计算[ab]的可超负荷算子
vertcat	用于计算[a;b]的可超负荷算子
le	用于计算 a<=b 的可超负荷算子
lt	用于计算 a<b 的可超负荷算子
gt	用于计算 a>b 的可超负荷算子
ge	用于计算 a>=b 的可超负荷算子
eq	用于计算 a==b 的可超负荷算子
ne	用于计算 a~=b 的可超负荷算子
not	用于计算~a 的可超负荷算子
and	用于计算 a&b 的可超负荷算子
or	用于计算 a|b 的可超负荷算子
subsasgn	用于计算 a(i)=b,a{i}=b,及 a.field=b 的可超负荷算子
subsref	用于计算 a(i),a{i},及 a.field 的可超负荷算子
colon	用于计算 a:b 的可超负荷算子
transpose	用于计算 a.'的可超负荷算子
ctranspose	用于计算 a'的可超负荷算子
subsindex	用于计算 x(a)的可超负荷算子

10. 数据分析与傅里叶变换函数

基本操作	
max	求最大值
min	求最小值
mean	求平均值
median	求中值
std	求标准差
sort	排序操作
sortrows	对矩阵的行按升序排序
sum	求各个元素之和
prod	求各个元素之积
hist	绘制柱状图
trapz	利用梯形法计算数值积分
cumsum	求各个元素的累积和
cumprod	求各个元素的累积积
cumtrapz	累积梯形法数值积分
有限差分	
diff	计算差分与近似计算
gradient	计算近似梯度
del2	离散拉普拉斯变换

续表

相关计算	
corrcoef	求相关系数
cov	求协方差矩阵
subspace	求子空间之间的夹角
滤波与卷积	
filter	一维数字滤波器
filter2	二维数字滤波器
conv	卷积与多项式乘积
conv2	二维卷积
convn	N 维卷积
deconv	反卷积与多项式除法
傅里叶变换	
fft	离散傅里叶变换
fft2	二维离散傅里叶变换
fftn	N 维离散傅里叶变换
ifft	离散傅里叶逆变换
ifft2	二维离散傅里叶逆变换
ifftn	N 维离散傅里叶逆变换
fftshift	将零点平移到频谱中心
声音和音频	
sound	将向量变为音频信号
soundsc	自动声音向量演奏
speak	将字符串变换为音频(仅用于 Macintosh 机)
recordsound	录音(仅用于 Macintosh 机)
soundcap	音频容量(仅用于 Macintosh 机)
mu2lin	将 mu- 方法编码的信号转换为线性信号
lin2mu	将线性信号转换为用 mu- 方法编码的信号
音频文件输入输出	
auwrite	写 NeXT/SUN(".a")音频文件
auread	读 NeXT/SUN(".au")音频文件
wavwrite	写微软 WAVE(".wav")音频文件
wavread	读微软 WAVE(".wav")音频文件
readsnd	读 SND 资源和文件(仅用于 Macintosh 机)
writesnd	写 SND 资源和文件(仅用于 Macintosh 机)
旧版本函数	
saxis	音频坐标轴刻度标记
实用工具	
playsnd	音频的实现程序

11. 泛函及 ODE 求解程序

最优化及方程求根	
fmin	单变量函数的极小化
fmins	多变量函数的极小化
fzero	寻找单变量函数的零点
数值积分	
quad	低阶法计算数值积分
quad8	高阶法计算数值积分
dblquad	数值评估双积分
绘 图	
ezplot	简单使用函数绘图仪
fplot	函数绘图
INLINE 函数对象	
inline	构造 INLINE 函数对象
argnames	变元名称
formula	函数形式
char	将 INLINE 对象转变为字符数组
实用工具	
vectorize	将字符串表达式或 INLINE 函数对象向量化
常微分方程求解	
ode45	高阶法求解常微分方程
ode23	低阶法求解常微分方程
ode113	求解非 stiff 常微分方程
ode15s	求解 stiff 常微分方程
ode23s	求解 stiff 常微分方程(低阶方法)
odefile	ODE 文件句法
ODE 选项处理	
odeset	生成或改变 ODE 选项结构
odeget	获取 ODE 选项参数
ODE 输出函数	
odeplot	时间序列 ODE 输出函数
odephas2	二维相平面 ODE 输出函数
odephas3	三维相平面 ODE 输出函数
odeprint	命令窗口 ODE 输出函数
帮助函数	
foptions	设置最优化程序的缺省参数
fcnchk	检查泛函的变元
innerlp	评价积分内循环
用于 ODE 求解程序的数字 Jacobian 矩阵帮助函数	
numjac	函数 F(t,y)的 Jacobian dF/dY 数学计算
colgroup	稀疏 Jacobian 矩阵的帮助函数

续表

用于 ODE 求解程序的事件定位及输出帮助函数	
odezero	以时间步长定位事件函数的任何零交叉
ntrp45	用于 ODE45 的插值帮助函数
ntrp15s	用于 ODE15S 的插值帮助函数
ntrp23	用于 ODE23 的插值帮助函数
ntrp23s	用于 ODE23S 的插值帮助函数
ntrp113	用于 ODE113 的插值帮助函数

12. 插值运算与多项式

数据插值	
interp1	一维数据插值
interp1q	快速一维线性插值
interpft	利用 FFT 进行一维数据插值
interp2	二维数据插值
interp3	三维数据插值
interpn	N 维数据插值
griddata	数据网络
样条插值	
spline	3 次样条数据插值
ppval	分段多项式计算
解析几何	
delaunay	Delaunay 三角形
dsearch	搜索最近点的 Delaunay 三角形
tsearch	最近三角形搜索
convhull	凸性分析
voronoi	Voronoi 图
inpolygon	是多边形区域内的点时返回真
rectint	矩形交叉区域
polyarea	生成多边形区域
多项式计算	
roots	求多项式根
poly	构造具有指定根的多项式
polyval	多项式计算
polyvalm	带矩阵变量的多项式的计算
residue	部分分式展开(留数计算)
polyfit	数据的多项式拟合
polyder	微分多项式
conv	多项式乘法
deconv	多项式除法

续表

实用工具	
xychk	检查一维和二维数据程序的变元
xyzchk	检查三维数据程序的变元
xyzvchk	检查三维体积数据程序的变元
automesh	如果输入数据能够自动网格化则返回真
mkpp	生成分段多项式
unmkpp	提供分段多项式的详细情况
resi2	重复极点的余数
tzero	零点转移
abcdchk	检查状态方程 ***A***,***B***,***C***,***D*** 矩阵的一致性
ss2tf	将状态空间系统变换为传递函数系统
ss2zp	将状态空间系统变换为零点-极点系统
tf2ss	将传递函数系统变换为状态空间系统
tf2zp	将传递函数系统变换为零点-极点系统
tfchk	传递函数正确性检查
zp2ss	将零点-极点系统变换为状态空间系统
zp2tf	将零点-极点系统变换为传递函数系统
旧版本函数	
icubic	一维三次插值
interp4	二维双线性数据插值
interp5	二维双三次数据插值
interp6	二维邻近数据插值
table1	一维表查找
table2	二维表查找

13. 字符串处理函数

一般函数	
char	建立字符串数组
double	将字符串转换为数值
cellstr	由字符数组建立字符串单元数组
blanks	建立空字符串
deblank	删除尾部的空字符串
eval	执行由 MATLAB 语句组成的字符串
字符串测试函数	
ischar	当变量为字符串时返回真
iscellstr	当变量为字符串单元数组时返回真
isletter	当变量为字母时返回真
isspace	当变量为空白时返回真
字符串比较函数	
strcat	字符串联结

续表

strvcat	字符串垂直联结
strcmp	比较两个字符串是否相同
strncmp	比较两个字符串的前 n 个字符是否相同
findstr	在一个字符串中查找另一个字符串
strjust	字母数组正确性检查
Strmatch	查找匹配的字符串
Strrep	字符串替换
Strtok	在字符串中查找标记
upper	将字符串转化成大写字母形式
lower	将字符串转化成小写字母形式
字符串与数值之间的转换函数	
num2str	将数值转换为字符串
int2str	将整数转换成字符串
mat2str	将矩阵转换成可用 eval 命令执行的字符串
str2num	将字符串转换成数值
sprintf	将数值变为格式控制下的字符串
sscanf	将字符串变为格式控制下的数值
基本数制之间的转换函数	
hex2num	将十六进制数转换为 IEEE 标准下的浮点数
hex2dec	将十六进制数转换为十进制数
dec2hex	将十进制数转换为十六进制数
bin2dec	将二进制数转换为十进制数
dec2bin	将十进制数转换为二进制数
base2dec	将以 B 为基的字符串转换为对应的十进制整数
dec2base	将十进制整数转换为以 B 为基的字符串
实用工具函数	
strings	关于字符串的帮助
旧版本函数	
str2mat	从各个字符串中形成填补有空白的字符矩阵
isstr	当变量为字符串时返回真
setstr	将数制转换为字符串

14. 时间与日期函数

当前的时间与日期	
now	以日期数字表示的当前日期与时间
date	以日期字符串表示的当前日期
clock	以日期向量表示的当前日期与时间
基本函数	
datenum	日期数字
datestr	日期字符串

续表

datevec	日期向量
日期函数	
calendar	日历
weekday	星期
eomday	月末
datetick	日期的格式化标记
时间函数	
Cputime	CPU 时间(以秒为单位)
tic,toc	秒表计时
etime	计时函数
pause	暂停

15. 通用图形函数

图形窗口的建立与控制	
figure	建立一个图形窗口
gcf	获取当前图形窗口的句柄
clf	清除当前图形窗口
shg	显示一个图形窗口
close	关闭一个图形窗口
refresh	刷新当前图形窗口
坐标系的建立与控制	
subplot	在指定位置建立坐标系
axes	在任意位置建立坐标系
gca	获取当前坐标轴的句柄
cla	清除当前坐标系
axis	坐标轴刻度及外观的控制
box	坐标轴方框的有无
caxis	控制伪色彩坐标轴刻度
hold	保持当前图形
ishold	是保持当前图形状态时返回真
图形对象的处理	
figure	建立一个图形窗口
axes	建立一个坐标系
line	建立一个直线对象
text	建立一个静态字符串对象
patch	建立一个多边形区域对象
surface	建立一个曲面对象
image	建立一个图像对象
light	建立一个光线
uicontrol	建立一个用户接口控件
uimenu	建立一个菜单

续表

图形句柄的操作	
set	设置图形对象的属性
get	获取图形对象的属性
reset	重置图形对象属性
delete	删除图形对象
gco	获取当前图形对象的句柄
gcbo	获取当前回调对象的句柄
gcbf	获取当前回调图形的句柄
drawnow	刷新未完成回头事件
findobj	搜索具有指定属性值的图形对象
copyobj	拷贝图形对象及其子对象
图形的硬拷贝与打印	
print	打印图形或保存图形到 M 文件
printopt	配置打印机缺省值
orient	设置打印纸方向
实用工具	
closereq	关闭图形窗口请求函数
newplot	NextPlot 属性的 M 文件之文件头
ishandle	是图形对象时返回真
实用打印工具	
bwcontr	调整黑白颜色对比度
hardcopy	将图形窗口保存到文件中
nodither	修正图形以避免抖动线
savtoner	修正图形以保存打印机调色效果
旧版本函数	
clg	清除图形窗口

16. 二维图形函数

基本二维图形	
plot	绘制线性图形
loglog	绘制对数坐标图形
semilogx	绘制半对数坐标图形(x 坐标为半对数坐标)
semilogy	绘制半对数坐标图形(y 坐标为半对数坐标)
polar	绘制极坐标图形
plotyy	绘制左右带有 Y 向标记字符串的图形
坐标系控制	
axis	坐标轴刻度及外观的控制
zoom	二维图形的放大及缩小
grid	设置网格线
box	坐标轴方框的有无

续表

hold	保持当前图形
axes	在任意位置建立坐标系
subplot	在指定位置建立坐标系
	图形注释
legend	图例说明
title	图形标题说明
xlabel	X 轴标记
ylabel	Y 轴标记
text	文本注释
gtext	用鼠标放置文本
	图形的硬拷贝及打印
print	打印图形或保存图形到 M 文件
printopt	配置打印机缺省值
orient	设置打印纸方向
	实用工具
lscan	合适的图例放置位置的扫描
moveaxis	获取及移动图例坐标轴

17. 三维图形函数

	基本三维图形
plot3	绘制三维空间的线和点
mesh	绘制三维网格曲面
surf	绘制三维彩色曲面
fill3	填充三维多面体
	颜色控制
colormap	设置颜色对照表
caxis	控制伪色彩坐标轴刻度
shading	设置颜色阴影模式
hidden	设置消影方式
brighten	色彩的明暗处理
	光照处理
surfl	绘制具有光照效果的三维阴影曲面
lighting	光照模式设置
material	设置材料的反光模式
specular	镜面反光
diffuse	漫射反光
surfnorm	求曲面法线
	颜色对照表的处理
hsv	色彩饱和值颜色对照表
hot	黑红黄白颜色对照表

续 表

gray	线性灰度颜色对照表
bone	带有淡兰色的灰度颜色对照表
copper	线性青铜色颜色对照表
pink	淡粉红色颜色对照表
white	全白色颜色对照表
flag	红白蓝黑交互色颜色对照表
lines	直线颜色的颜色对照表
colorcube	增强型颜色立方的颜色对照表
jet	HSV 色彩模型变量
prism	棱镜颜色对照表
cool	青色和洋红色阴影颜色对照表
autumn	红色和黄色阴影颜色对照表
spring	洋红色和黄色阴影颜色对照表
winter	蓝色和绿色阴影颜色对照表
summer	绿色和黄色阴影颜色对照表
坐标系控制	
axis	坐标轴刻度及外观的控制
zoom	图形的放大及缩小
grid	设置网格线
box	坐标轴方框的有无
hold	保持当前图形
axes	在任意位置建立坐标系
subplot	在指定位置建立坐标系
视点控制	
view	指定三维图形的视点
viewmtx	视点变换矩阵
rotate3d	交互式旋转三维图形的视点
图形注释	
title	图形标题说明
xlabel	X 轴标记
ylabel	Y 轴标记
zlabel	Z 轴标记
colorbar	显示彩色刻度条
text	文本注释
gtext	用鼠标放置文本
图形的硬拷贝及打印	
print	打印图形或保存图形到 M 文件
printopt	配置打印机缺省值
orient	设置打印纸方向

18. 特殊图形函数

特殊二维图形的绘制	
area	绘制区域填充图
bar	绘制垂直放置的直方图
barh	绘制水平放置的直方图
bar3	绘制垂直放置的三维直方图
bar3h	绘制水平放置的三维直方图
comet	绘制轨迹线
errorbar	绘制条形误差图
ezplot	简单使用函数绘图仪
feather	绘制复数向量
fill	填充二维多边形
fplot	函数图形绘制
hist	绘制概率分布图
pareto	绘制 Pareto 图
pie	绘制圆饼图
pie3	绘制三维圆饼图
plotmatrix	矩阵的分散绘制
ribbon	绘制二维带状图
stem	离散序列图绘制
stairs	阶梯图绘制
等高线及二维半图形的绘制	
contour	等高线绘制
contourf	填充等高线的绘制
contour3	三维等高线的绘制
clabel	等高线高度标记
pcolor	伪色彩绘制命令
quiver	绘制梯度场
voronoi	绘制 Voronoi 图
特殊三维图形的绘制	
comet3	绘制三维轨迹线
meshc	绘制带有等高线的网格曲面
meshz	绘制带有底座的网格曲面
stem3	三维离散序列图绘制
quiver3	绘制三维梯度场
slice	绘制体积切片图
surfc	绘制带有等高线的阴影曲面
trisurf	绘制由三角曲面片构成的阴影曲面
trimesh	绘制由三角曲面片构成的网格阴影曲面
waterfall	绘制瀑布状网格曲面图
图像显示及其文件输入输出	
image	显示一幅图像

续表

imagesc	数据刻度定标及其图像显示
colormap	设置颜色对照表
gray	线性灰度颜色对照表
contrast	颜色对照表灰度化以增强图像对比度
brighten	颜色的明暗处理
colorbar	显示色彩刻度条
imread	从图像文件读取图像
imwrite	将图像写入图像文件
imfinfo	获取图像文件的相关信息
动画技术	
capture	当前图像的屏幕抓取
moviein	影片帧初始化
getframe	获取影片帧
movie	放映已记录的影片帧
qtwrite	将影片转换为快速格式(仅用于 Macintosh 机)
rotate	按指定的点和方向旋转对象
frame2im	将影片帧转换为有索引的图像
im2frame	将有索引的图像转换为影片帧
与颜色相关的函数	
spinmap	旋转颜色对照表
rgbplot	绘制颜色对照表
colstyle	由字符串分离颜色及线型样式
实体造型	
cylinder	绘制圆柱体
sphere	绘制球体
patch	绘制多边形区域
实用工具	
makebars	生成绘制直方图的数据
contourc	等高线计算
contours	非矩形曲面的等高线
旧版本函数	
compass	复数向量绘制
rose	绘制玫瑰线

19. 文件输入输出函数

打开与关闭文件	
fopen	打开一个文件
fclose	关闭一个文件
二进制文件的输入与输出操作	
fread	从文件中读取二进制数据
fwrite	将二进制数据写入到文件中

续表

文件的格式化输入与输出操作	
fscanf	以指定格式从文件中读取数据
fprintf	将数据以指定格式写入文件
fgetl	从文件中读取一行数据(去掉换行符号)
fgets	从文件中读取一行数据(保留换行符号)
input	提示用户输入数据
字符串变换	
sprintf	将数据以指定格式写入一个字符串
sscanf	以指定格式从一个字符串中读取数据
文件指针定位	
ferror	查询文件的错误状态
feof	文件末尾测试
fseek	设置文件指针位置
ftell	获取文件指针位置
frewind	重置文件指针位置到文件头
文件名处理	
matlabroot	安装 MATLAB 的根目录
filesep	从完整的文件名中剥离目录名
pathsep	从完整的文件名中剥离路径名
mexext	MEX 文件名扩展
fullfile	建立一个完整的文件名
partialpath	部分路径名
tempdir	获取临时目录
tempname	获取临时文件
文件输入输出函数	
load	从 MAT 文件调入变量到工作空间
save	将工作空间的变量存入到 MAT 文件
dlmread	读 ASCII 文件
dlmwrite	写 ASCII 文件
wk1read	读电子数据表(WK1)文件
wk1write	写电子数据表(WK1)文件
图像文件输入输出函数	
imread	从图像文件读取图像
imwrite	将图像写入图像文件
imfinfo	获取图像文件的相关信息
音频文件输入输出函数	
auwrite	写 NeXT/SUN(". au")音频文件
auread	读 NeXT/SUN(". au")音频文件
wavwrite	写微软 WAVE(". wav")音频文件
wavread	读微软 WAVE(". wav")音频文件
命令窗口输入输出	
clc	清除命令窗口

续表

home	光标回原点
disp	显示数组
input	提示用户输入
pause	暂停
实用工具	
str2rng	将电子数据表中的字符串转换为数值数组
wk1const	定义 WK1 记录类型
wk1wrec	写 WK1 记录头
hdf	MEX 文件到 HDF 库的接口
旧版本函数	
csvread	读用逗号分隔的格式化数值数据文件
csvwrite	写用逗号分隔的格式化数值数据文件

20. 动态数据交换

动态数据交换函数	
ddeadv	建立交换链
ddeexec	发送要执行的字符串
ddeinit	初始化 DDE 会话过程
ddepoke	向应用程序发送数据
ddereq	从应用程序请求数据
ddeterm	终止 DDE 会话过程
ddeunadv	释放交换链

21. 图形用户接口工具

GUI 函数	
uicontrol	建立用户接口控件
uimenu	建立用户接口菜单
ginput	用鼠标读取图形数据点
dragrect	用鼠标拖拉一个异或矩形
rbbox	拉橡皮框
selectmoveresizeobjects.	交互式选择、移动、改变大小、拷贝处理
waitforbuttonpress	等待按键或击命令按钮
waitfor	阻塞执行过程并等待事件
uiwait	阻塞执行过程并等待重新执行
uiresume	重新执行阻塞了的 M 文件
GUI 设计工具	
guide	GUI 的可视化设计工具

续表

align	排列控件及坐标轴
cbedit	编辑回调函数
menuedit	编辑设计菜单
propedit	属性编辑
对话框	
dialog	建立一个对话框
axlimdlg	坐标轴范围对话框
errordlg	错误信息对话框
helpdlg	帮助信息对话框
inputdlg	数据输入对话框
listdlg	列表选择对话框
menu	生成用户菜单
msgbox	消息显示对话框
questdlg	提问对话框
warndlg	警告信息对话框
uigetfile	标准的打开文件对话框
uiputfile	标准的保存文件对话框
Uisetcolor	标准的颜色选择对话框
uisetfont	标准的字体选择对话框
pagedlg	标准的页面位置对话框
printdlg	打印对话框
waitbar	显示等待条
菜单实用工具	
makemenu	创建菜单结构
menubar	依赖于机器的菜单条缺省属性设置
umtoggle	设置菜单项目的选中标记
winmenu	创建“Window”菜单项目的子菜单
工具条按钮组实用工具	
btngroup	创建工具条按钮组
btnstate	查询工具条按钮组的状态
btnpress	工具条按钮组的压键管理函数
btndown	按下工具条按钮组的一个按钮
btnup	抬起工具条按钮组的一个按钮
图形及坐标系的用户定义属性之实用工具	
clruprop	清除用户定义属性
getuprop	获取用户定义属性的属性值
setuprop	设置用户定义属性的属性值
多功能实用工具	
allchild	获取所有的子对象
findall	搜索所有的对象
hidegui	隐藏或不隐藏 GUI
edtext	坐标轴文字对象的交互式编辑
getstatus	获取图形窗口中文字串的状态

续表

setstatus	设置图形窗口中文字串的状态
popupstr	获取从下拉式菜单中所选择的字符串
remapfig	变换图形对象的位置
setptr	设置图形指针
getptr	获取图形指针
overobj	获取图形指针所在位置的对象的句柄
实用工具	
textwrap	返回给定用户控件的包装字符串矩阵
btnicon	用于 BTNGROUP 的图标库
icondisp	显示 BTNGROUP 的第 n 个图标
ctlpanel	GUIDE 控制面板的初始化
fignamer	选择下一个可用图形的名称
tabdlg	创建和管理对话框
初始函数	
hthelp	MATLAB 帮助文件的超文本帮助工具
htpp	HTHELP 的超文本处理器
loadhtml	HTHELP 的 HTML 分离
matqdlg	工作空间变换对话框
layout	定义对话框布置参数的程序
menulabel	分离用于键盘操作和快捷键的菜单说明标记
extent	获取独立的字符串扩展属性
figflag	如果图形当前被显示在屏幕上时返回真
fwhich	在 MATLAB 路径中定位文件及目录
isdir	如果变元是一个目录时返回真
matqueue	创建并操作一个基于图形的队列
matq2ws	工作空间变换对话框(MatqDlg)的帮助工具
ws2matq	工作空间变换对话框(MatqDlg)的帮助工具
matqparse	工作空间变换对话框(MatqDlg)的对话框入口分离工具
旧版本函数	
printmenu	在命令窗口中打印下拉式菜单

参考文献

1 张培强. MATLAB 语言——演草纸式的科学计算语言. 合肥:中国科技大学出版社,1995

2 李宜达. 动态模拟与绘图——使用 MATLAB/SIMULINK. 台北:全华科技图书股份有限公司,1998

3 薛定宇. 控制系统计算机辅助设计——MATLAB 语言及应用. 北京:清华大学出版社,1996

4 楼顺天,于卫,闫华梁. MATLAB 程序设计语言. 西安:西安电子科技大学出版社,1997

5 施阳. MATLAB 语言精要及动态仿真工具 SIMULINK. 西安:西北工业大学出版社,1997

6 施阳. MATLAB 语言工具箱——TOOLBOX 实用指南. 西安:西北工业大学出版社,1998